Image and Texture Analysis using Biorthogonal Angular Filter Banks

A Thesis
Presented to
The Academic Faculty

by

Jose Gerardo Gonzalez Rosiles

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
July 2004

Image and Texture Analysis using Biorthogonal Angular Filter Banks

Approved by:

Dr. Russell M. Mersereau, Committee Chair	Dr. James H. McClellan
Dr. Mark J. T. Smith, Advisor	Dr. A. P. Sakis Meliopoulos
Dr. Mark H. Clements	Dr. Christopher Heil School of Mathematics

Date Approved: July 6, 2004

To my mother Lucila for her unconditional love,

sacrifice, hard work and strength.

To my brother Cesar for his support and friendship.

To my daughter Paulina for the happiness she has brought to our life.

To my wife,

Laura,

for her infinite support, patience and love...

without you this work would have not been possible.

Thanks for sharing this dream.

ACKNOWLEDGEMENTS

This has been a long journey for me. Over the years I met many great people that have contributed in many ways to the completion of my Ph.D. At the end I will have more than a title and a diploma, I will have many years of experiences and life lessons that will allow me to move forward with a firmer walk, and as stronger and better individual. There are a few persons that I would like to extend a personal thank you.

I would like to start by thanking the members of my thesis committee. To Dr. James McClellan and Dr. Mark Clements, I appreciate the feedback and interest provided as readers of this dissertation. To Dr. Russell Mersereau, I am deeply grateful for taking the extra duties of serving as a co-advisor. I would like to thank Dr. Christoper Heil and Dr. Sakis Meliopoulos for taking time of their schedule to attend my defense presentation.

To my advisor Dr. Mark J. T. Smith I am infinitely indebted for his support (moral and financial), guidance and literally infinite patience in the completion of this thesis. I am also grateful for the many professional and ethical advise he shared with me over the years.

I would like to thank the staff at CSIP, in particular Kay Gilstrap, Stacy Schultz, and Charlote Dougthy who helped me with so many administrative issues during my residence at CSIP. I thank Christy Ellis who took on many extra tasks at the end as I was physically away from CSIP.

To all of the CSIP students with whom I shared illuminating discussions, a game of soccer, a lunch, and many laughs, I give them my sincerest thanks. In particular I thank Alen Docef, Hong Man, Paul Hong, Sang Park, Asmat Khan, Doug Britton, Joe Monaco, Qin Jiang, and Joey Arrowood for their fellowship. To my friends Greg and Beth Showman I thank them for the many dinners they shared with me, and to Greg for the many great masses we played together at the Catholic Center. To my partner in crime Tami Randolph, even though she is difficult, I hope our friendship lasts forever.

This work could not be explained without all the hands that were carrying me and

pushing me over the years. First I want to thank my mother for all her unconditional sacrifices, for the many long hours of hard work she has put over her life so that her sons would have a better opportunity at life. Mom, I hope I have full filled your dream. To my brother I am grateful that he took on being the "man of the house" when I left for this adventure. I would have never done it as well as you have, thanks for taking care of our mother and our home.

To my wife Laura, there is so much I could say that I could fill a few pages. This dissertation would have never been completed without her support and patience, ... and lots of pushing. Thanks for listening for hours and hours to uncomprehensible technical concepts. Thanks for sharing this dream, and at times making it more yours than mine. Thanks for all the free time you didn't have with me because I was in the other room "working on my thesis." Thanks for all your unconditional love, I can only hope I can give you the same back to you and our family.

Finally, I thank God for giving me the strength, faith, and hope to keep going. I thank God for my friends and my family who never gave up on their support on this dream. I thank God for giving me the chance and the life to reach this moment. I thank God for all the blessings I have received over these years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In this thesis we develop algorithms for the processing of textures and images using a ladder-based biorthogonal directional filter bank (DFB). This work is based on the DFB originally proposed by Bamberger and Smith. First we present a novel implementation of this filter bank using ladder structures. This new DFB provides non-trivial FIR perfect reconstruction systems which are computationally very efficient. Furthermore we address the lack of shift-invariance in the DFB by presenting a novel undecimated DFB that preserves the computational simplicity of its maximally decimated counterpart. Finally, we study the use of the DFB in combination with pyramidal structures to form polar-separable image decompositions.

Using the proposed filter banks we develop and evaluate algorithms for texture classification, segmentation and synthesis. We perform a comparative study with other image representations and find that the DFB provides some of the best results reported on the data sets used.

Using the proposed directional pyramids we adapt wavelet thresholding algorithms. We find that our decompositions provide better edge and contour preservation than the best results reported using the undecimated discrete wavelet transform.

Finally, we apply these algorithms to the analysis and processing of synthetic aperture radar (SAR) imagery. SAR image analysis is impaired by the presence of speckle noise. First, we study the removal of speckle for visual enhancement of the image. Additionally, we implement land cover segmentation and classification algorithms taking advantage of the textural characteristics of SAR images. Finally, we propose a model-based SAR image compression algorithm where the speckle component is separated from the structural features of a scene. The speckle is captured with a texture model and the scene component is coded with a wavelet coder at very low bit rates. The resulting decompressed images have a better perceptual quality than SAR images compressed without removing speckle.

CHAPTER I

INTRODUCTION

In this thesis we develop algorithms for the processing of textures and images using a ladder-based biorthogonal directional filter bank (DFB). This work is based on the DFB originally proposed by Bamberger and Smith in [5, 8]. First we present a novel implementation of this filter bank using ladder structures. This contribution brings a new set of features that increases the utility of the DFB. This new DFB provides non-trivial FIR perfect reconstruction systems which are computationally very efficient. Moreover we inherit such features of ladder structures as integer-to-integer representations and directional representations over finite fields. Furthermore we address the lack of shift-invariance in the DFB by presenting a novel undecimated DFB that preserves the computational simplicity of its maximally decimated counterpart. Finally, we study the use of the DFB in combination with pyramidal structures.

One drawback of previous work on directional image processing is the high computational and storage cost. The DFB used in this thesis is maximally decimated and efficient to implement in the polyphase domain while achieving good frequency domain characteristics. Using the DFB, we explore different applications where the extraction of directional information has been shown to be useful. We aim to show that the DFB is one of the best tools reported to date for directional processing and analysis of images.

First the DFB is studied in the area of texture analysis and synthesis. It has been established that multichannel representations such as 2-D Gabor decompositions, discrete wavelet transforms (DWTs), steerable pyramids, etc., provide some of the best results for texture classification, segmentation and synthesis. It is then a natural step to explore the use of the DFB on these well known problems. We will show that the DFB is not only a viable alternative, but performs at the same level or better than other decompositions while maintaining low complexity.

Another problem that has been of recent interest is the non-linear estimation of signals contaminated by noise using wavelet domain thresholding. For the case of 2-D signals (like natural images), it has been observed that overcomplete representations like the undecimated 2-D DWT outperforms the 2-D DWT. Hence, we evaluate the representation of edge information by pyramidal decompositions based on the DFB and develop algorithms which improve the perceived visual quality.

The algorithms developed for texture analysis/synthesis and image denoising are applied in the third part of this research to the analysis and processing of synthetic aperture radar (SAR) imagery. SAR images, like other coherent imaging products, suffer from the presence of speckle, which reduces the ability to analyze and process the image. Our first objective will be to study the removal of speckle to enhance the visual quality of the image. Additionally, we exploit the textural characteristics of SAR images by implementing land cover segmentation and classification algorithms based on the texture analysis schemes developed in the fist part of the thesis. Finally, we propose a model-based compression algorithm in which the speckle component is separated from the structural features of a scene. The speckle can be considered a texture and is modelled with a compact set of parameters, while the features of the scene are retained for very low bit rates.

1.1 The Directional Filter Bank

We present the theory of the Directional Filter Bank (DFB) introduced by Bamberger [5], and Bamberger and Smith [8]. We assume some familiarity with the concepts of multirate multidimensional systems, and direct the reader to Chapter 12 of [130] and [133] for detailed reviews on this subject.

The DFB splits the frequency cell $[-\pi, \pi)^2$ into an even number of wedge-shaped regions as shown in Figure 2. Each wedge-shaped band captures image detail along orientations orthogonal to the subband direction. The obvious approach to implement the DFB is as a parallel M-channel structure. For example, the structure for an eight-band DFB is shown in Figure 1. Here the bands are maximally decimated by the matrices $\mathbf{D}_i$ ($|\det(\mathbf{D}_i)| = 8$). However, designing 2-D filters with the required passbands while achieving other properties

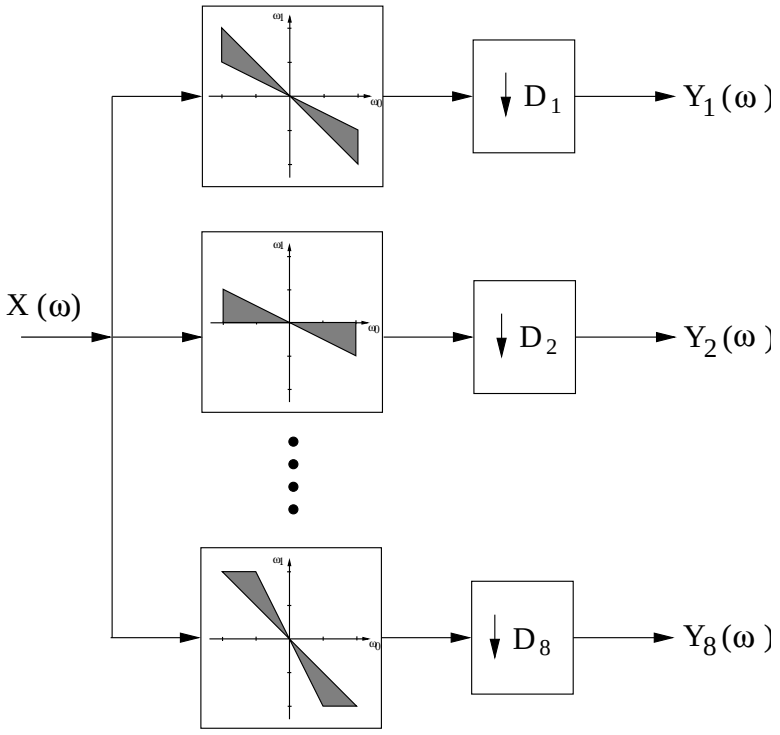


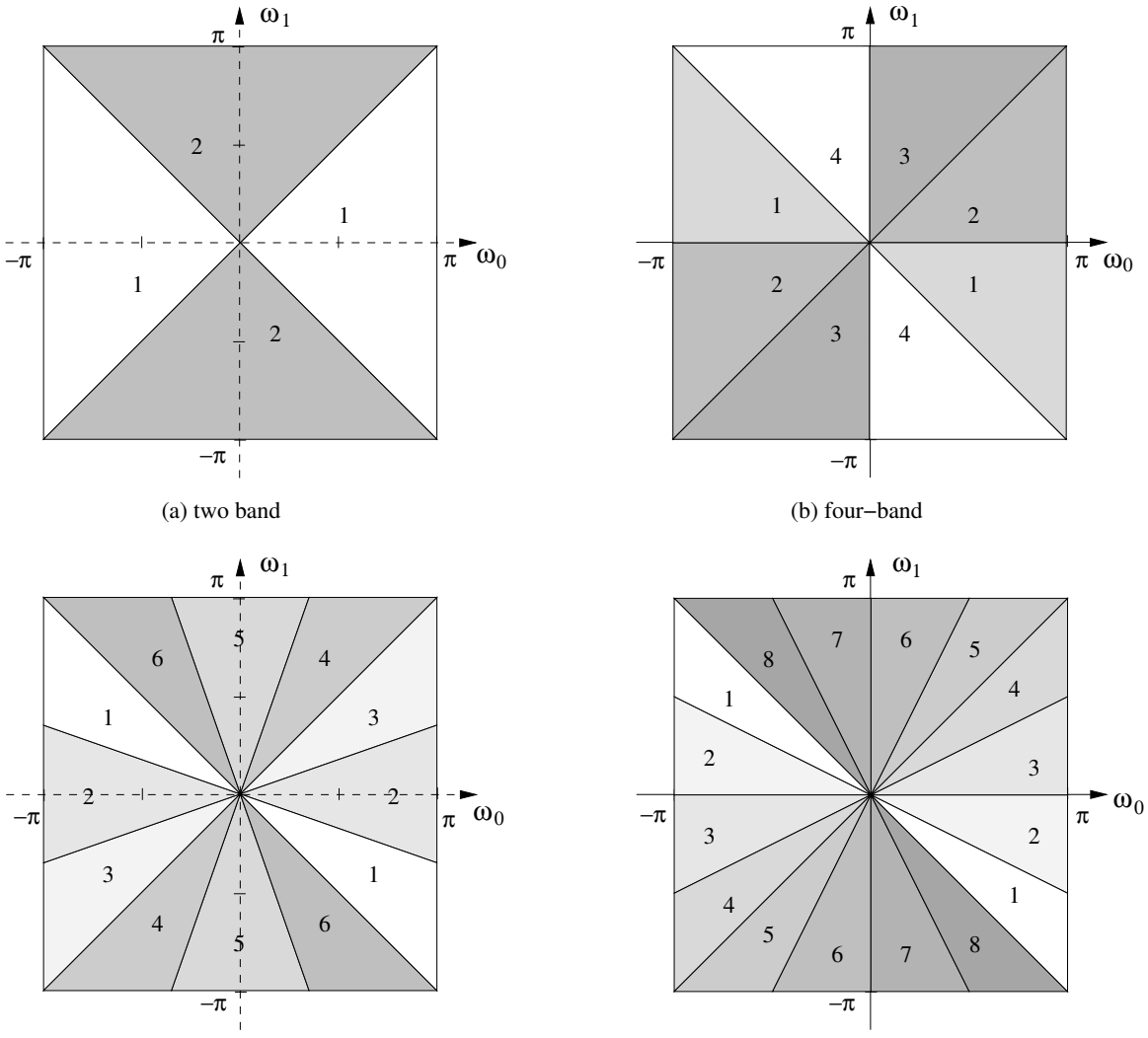
Figure 1: Parallel implementation of a maximally decimated eight-band DFB. Note that $|\det(\mathbf{D}_i)| = 8$

like alias cancellation and perfect reconstruction (PR) is a non-trivial task.

Bamberger resorted to the basic principle of multirate systems where simpler filter bank structures are cascaded to realize the desired passband. In this thesis we specifically adopt the uniform M-stage tree structure that generates $N = 2^M$ subbands. The tree structure partitioning can be illustrated graphically with Figure 2-(a),(b),(d), where the subbands of the two-band system can be further divided into a four-band decomposition, and a similar step can be taken to go from Figure 2-(b) to the eight-band plane partitioning of Figure 2-(d).

The basic element of Bamberger's DFB is the diamond filter bank shown in Figure 3. This 2-D filter bank consists of two complementary diamond-shaped filters followed by downsampling by a quincunx matrix , where

$$\mathbf{Q} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \tag{1}$$



(a) two band

(b) four−band

(c) six−band

(d) eight−band

Figure 2: Frequency band partitions achieved by the Bamberger DFB

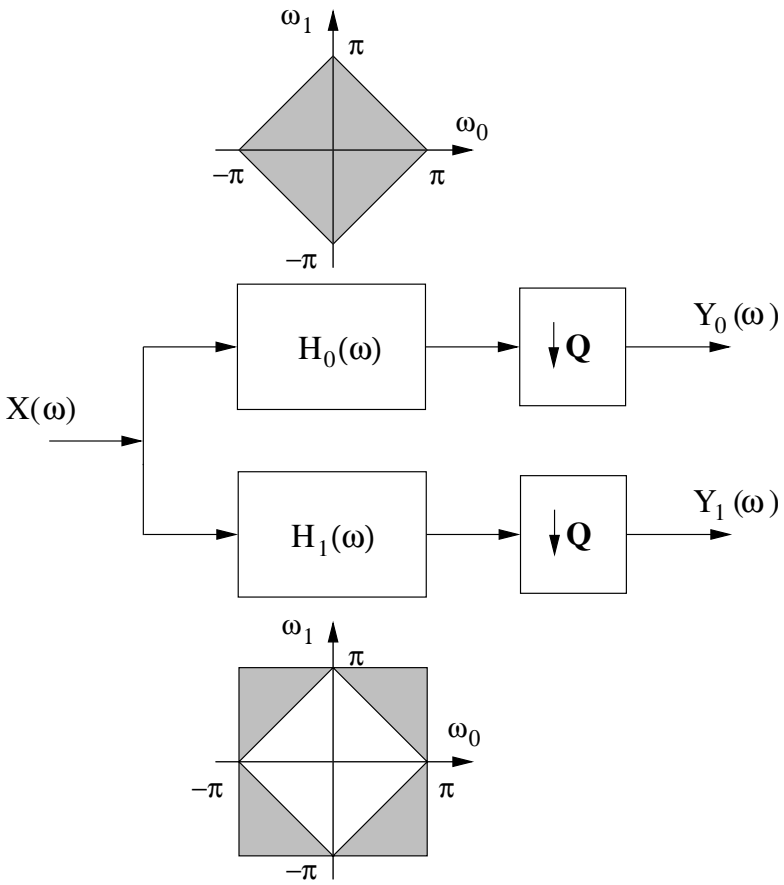


Figure 3: Block diagram for the diamond filter bank showing the ideal support for the 2-D filters.

The diamond filters provide alias free decimation for $\mathbf{Q}$. The two-channel diamond filter bank is presented in Figure 3.

The construction of tree-structured eight-band DFB using diamond filter banks is shown in Figure 4 which achieves the partitioning of Figure 2-(d).

Next, we describe in more detail the different stages of this DFB, but we refer the reader to [5] for a comprehensive discussion. Without loss of generality, we concentrate on describing the processing for the first channel of the tree structure which consists of the tree stages shown in Figure 5.

We let $\mathbf{Q}_j = \mathbf{Q}$ for $i = 1, 2, 3$, and choose unimodular matrix $\mathbf{U}_1$ as

$$\mathbf{U}_1 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

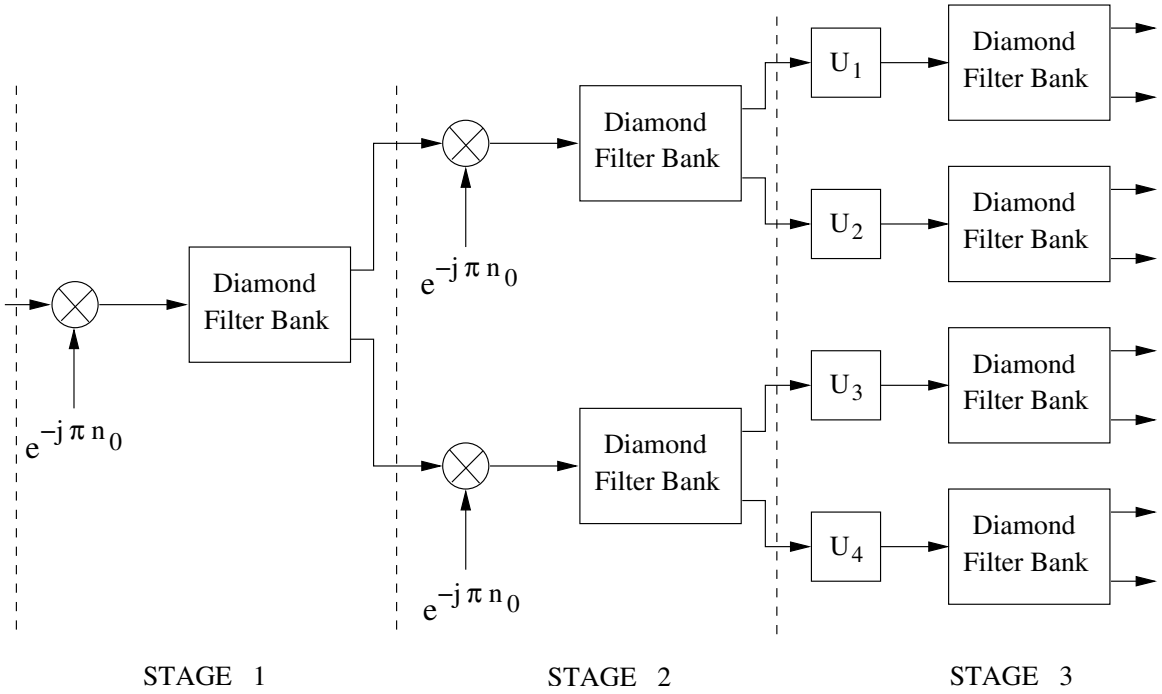


Figure 4: Block diagram for the eight-band tree structured Bamberger DFB.

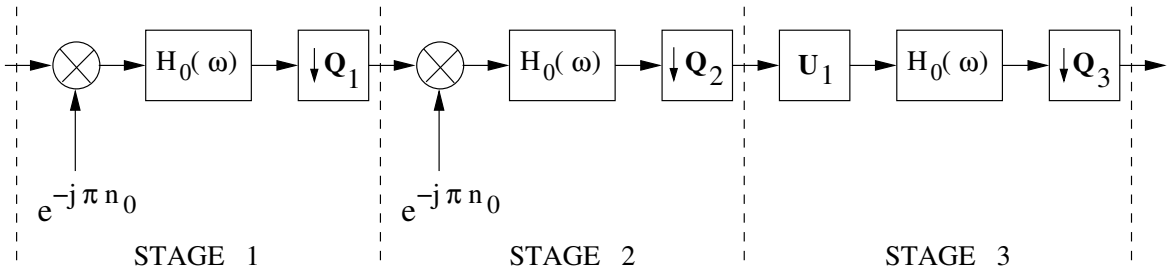


Figure 5: Diagram of the isolated tree branches for the first channel of a tree-structured eight-band DFB.

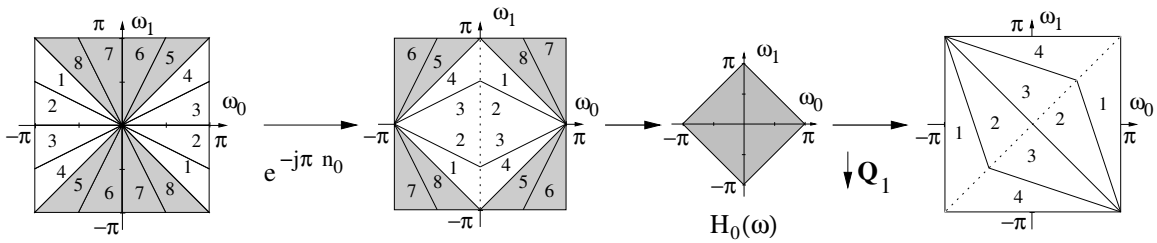


Figure 6: Frequency domain processing for the first stage of tree-structured Bamberger DFB (only the first channel is shown.)

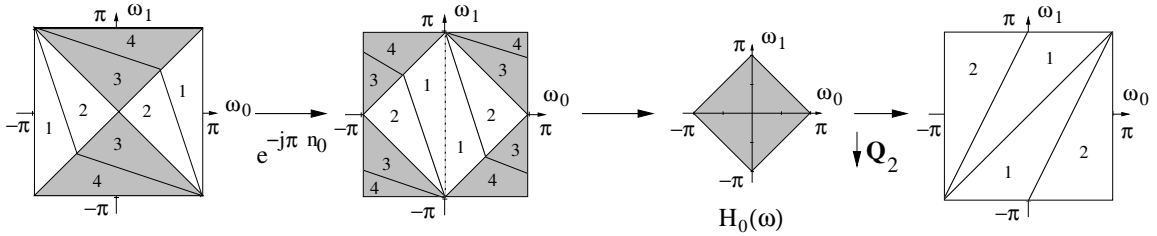


Figure 7: Frequency domain processing for the second stage of tree-structured Bamberger DFB (only the first channel is shown.)

according to the matrix selection rules derived by Park, et al., in [96]. We should note this quincunx matrix is one of many different possibilities. This matrix has been chosen because it inflicts a *clockwise* 45 degree rotation in the frequency domain, and it facilitates the explanation that follows. Each stage of the tree structure is explained next.

In Figure 6, we show the frequency domain processing for stage I. First, the modulation by the complex exponential generates a frequency shift by π in the ω_0 direction. This operation rearranges the location of subband subsets $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ so that they can be separated by the diamond filter bank. The downsampling step stretches and rotates the filter output to cover the full $[-\pi, \pi)^2$ area.

The processing on stage II is identical to that of stage I as shown in Figure 7. In this case subbands $\{1, 2\}$ are separated from subbands $\{3, 4\}$. Again, the subbands have been shuffled so that they are packed within the $[-\pi, \pi)^2$ cell.

The output of stage II delineates the wedge-shaped regions of Figure 2-(d) for bands 1 and 2. These regions have been rearranged to form complementary parallelogram regions.

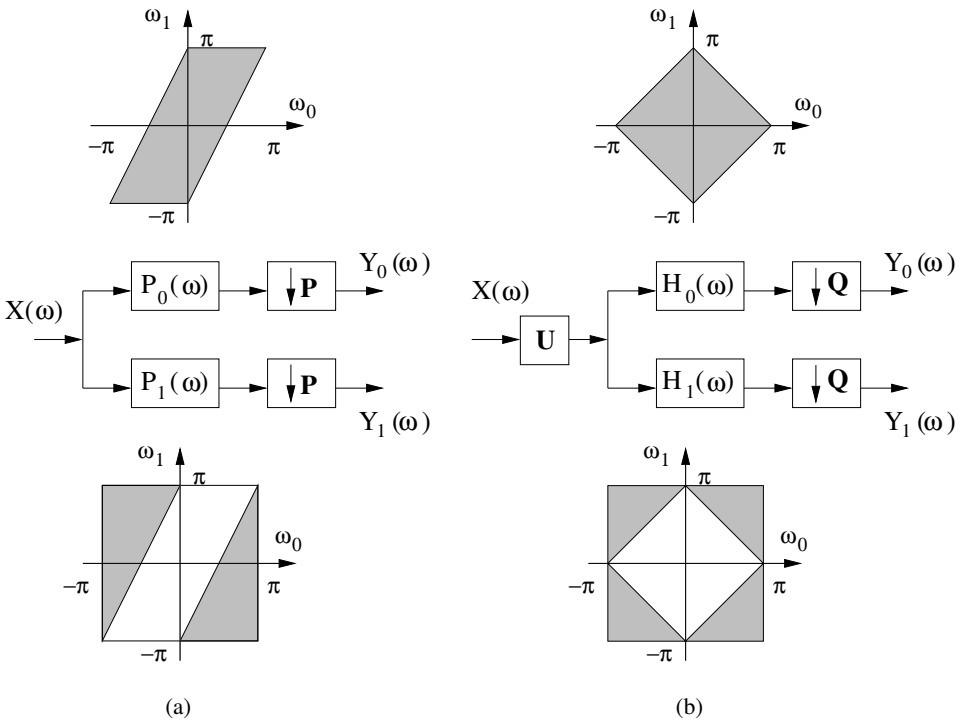


Figure 8: The parallelogram filter bank. (a) Two channel structure. (b) Equivalent implementation using the diamond filter bank with unimodular matrices.

Hence, we would need to use a two-channel filter bank with the appropriate parallelogram-shaped filters as shown in Figure 8-(a). The matrix $\mathbf{P}$ is such that alias free decimation is achieved with the parallelogram filters $P_0(\omega)$ and $P_1(\omega)$.

Fortunately, as described by Bamberger [5], we can take advantage of the properties of 2-D multirate systems. In Figure 4, we note the presence of matrices $\mathbf{U}_i$ for $i = 1, 2, 3, 4$ at the frontend of stage 3. These matrices are unimodular, and have the effect of mapping the parallelogram supports into diamond supports. Since $|det(\mathbf{U}_i)| = 1$, the sample density is preserved and only the subband samples are rearranged on the 2-D spatial plane. The equivalent system using diamond filters is depicted on Figure 8-(b). For a detailed derivation of the equivalence between the two systems in Figure 8 we refer the reader to Appendix A.

As derived in Appendix A, the matrix $\mathbf{U}_1$ is given by

$$\mathbf{U}_1 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \tag{2}$$

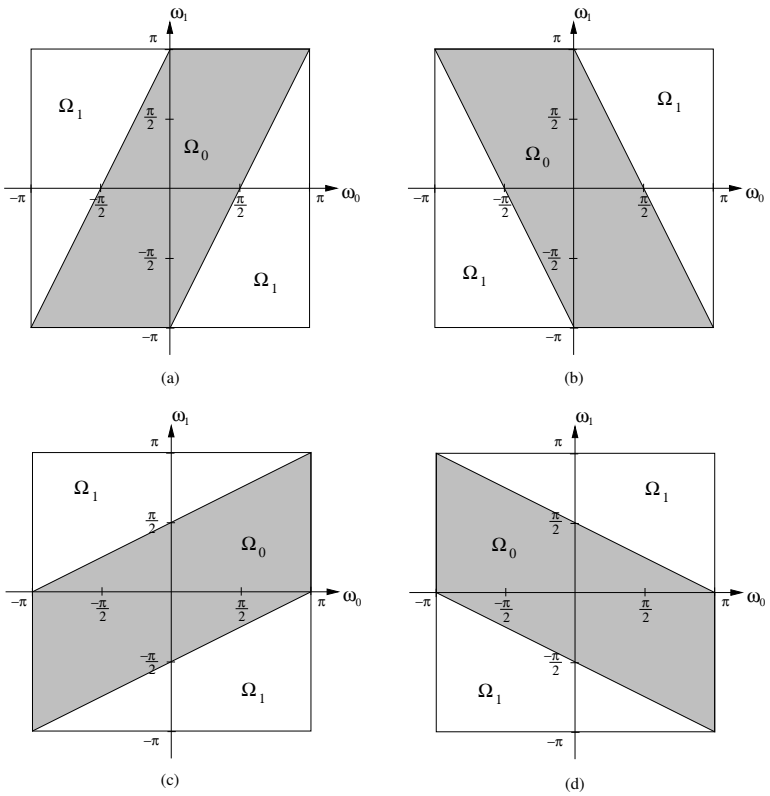


Figure 9: Possible parallelogram partitions of the frequency plane. These are used to determine the filters required for each channel on the third stage of the eight-band DFB

The remaining channels in the third stage of the DFB also need unimodular matrices $\mathbf{U}_i$. The four possible parallelogram supports are shown in Figure 9, implying a different unimodular matrix for each case. The remaining three matrices are obtained using the same derivation from Appendix A, and are given by

$$\mathbf{U}_2 = \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{U}_3 = \begin{bmatrix} 0 & 1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{U}_4 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}. \tag{3}$$

In the case of a tree structure with three or more stages, the proper quincunx and re-diamond matrices need to be selected for each stage. Rather than going through the painstaking process described in Appendix A, we refer the reader to the work by Park, et al. [96]. In this work, a set of rules to select these resampling matrices with ease has been derived.

Referring back to Figure 4, the processing of stage three is shown in Figure 10. Once

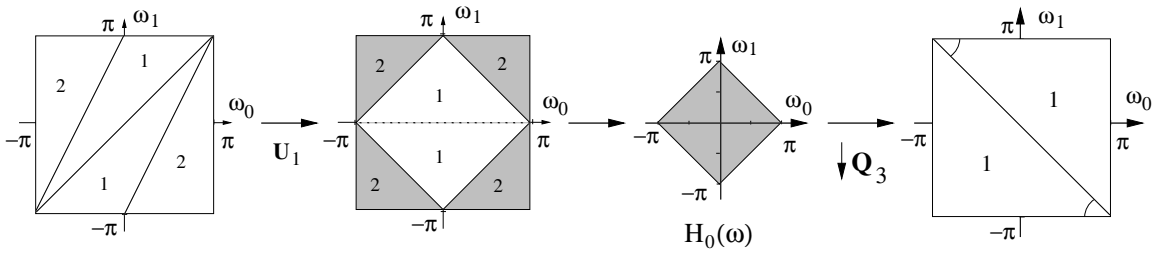


Figure 10: Frequency domain processing for the third stage of tree-structured Bamberger DFB (only the first channel is shown.)

the information has been rearranged into diamond-shaped regions by resampling with $\mathbf{U}_1$, then the processing is finished with the diamond filter bank as in the previous stages.

The tree-structured DFB we have described can be extended to any 2^M DFB by replicating the structures in stage 3 and selecting the proper unimodular and downsampling matrices (see [96]).

Although the synthesis section of the DFB has not been described, it should be clear that it consists of the reverse operations. In this case the $\mathbf{U}_i$ matrices have the reverse task of reshaping the subband information from diamond supports to parallelogram regions.

1.2 Improvements to the Bamberger DFB

Subband based image processing often requires the ability to establish local spatial relationships among the subband coefficients at the intra- and inter-subband levels, and/or calculation of statistics and energy-like measurements from the data. There are a couple of drawbacks that hindered such processing using the Bamberger DFB.

- First, the use of modulators to rearrange the data into a diamond support introduces severe spatial distortion (visual and quantitative) into the subbands by shuffling the low and high frequencies. This problem is compounded by the stretching and rotation induced by the decimation operations.

- Second, as will be shown, for 2^M-band DFBs with $M > 2$, each subband lies over a different lattice geometry. Therefore, this condition makes it difficult to establish inter-subband relationships or dependencies.

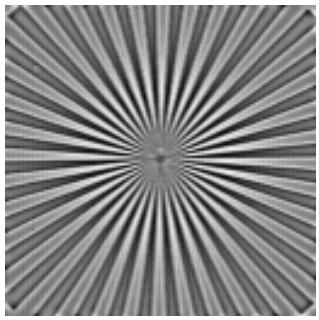


Figure 11: Test image.

In Figure 12 we illustrate the problems cited above by processing the test image of Figure 11. The actual design of the diamond-shaped filters is explained in the next section.

First we see a clear checkerboard effect caused by the modulation by π. Second we see the skewed placement of the subbands over the 2-D spatial plane, and how this placement is different on each subband.

It seems that the motivation to introduce the modulators in the tree structure was to contain/package the frequency domain expansions caused by quincunx downsampling to the $[-\pi, \pi)^2$ region. However, this is not a required condition to maintain alias free quincunx decimation. To avoid the use of modulators, Bamberger [6] proposed using the multirate identities from Figure 13 to migrate all the modulators to the right of the analysis tree structure in Figure 4, and to move all the filters to left of the structure. The opposite operations can be applied to the synthesis stage. This is equivalent to collapsing the tree-structure into the parallel structure of Figure 1. The analysis side modulators are cancelled with the synthesis side modulators. For instance, for an eight-band DFB, the overall analysis filters are given by

$$G_\ell(\omega) = F_j(\omega) F_k(\mathbf{Q}_1^T \omega) F_m(\mathbf{U}_i^T \mathbf{Q}_2^T \mathbf{Q}_1^T \omega) \tag{4}$$

where $j, k, m \epsilon \{0, 1\}$, $i \epsilon \{1, 2, 3, 4\}$, and $\ell = 1, 2, ...8$. Looking at the tree structure as a binary tree, we have $\ell = j + 2k + 4m + 1$. The filters $F_0(\omega)$ and $F_1(\omega)$ are obtained by a modulation

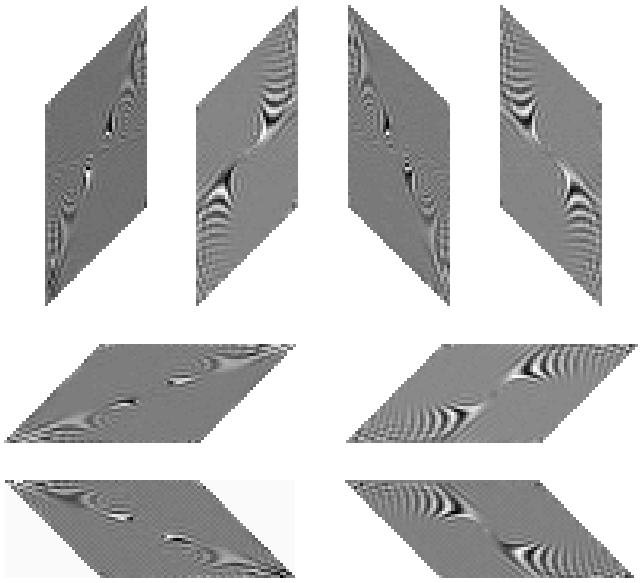


Figure 12: Output of the original Bamberger DFB showing the effects of modulation and non-rectangular sampling geometry.

operation (Figure 13-(c)) which in frequency domain is expressed by

$$\begin{aligned} F_0(\omega_0, \omega_1) &= H_0(\omega_0 - \pi, \omega_1) \\ F_1(\omega_0, \omega_1) &= H_1(\omega_0 - \pi, \omega_1). \end{aligned} \tag{5}$$

As $H_0(\omega)$ and $H_1(\omega)$ have diamond support, the modulation generates fan filters.

It was noted by Park, et al., [96] that the modulators can be eliminated as described by Bamberger while keeping the tree structure otherwise intact. Additionally, as shown in the next section keeping the tree structure allows for very efficient implementation of diamond and fan filter banks. Hence, in this thesis we assume that the tree-structured Bamberger DFB is implemented with fan filter banks (FFBs).

The FFB structure is shown in Figure 14 with the corresponding ideal filter supports. The complementary fan filters are denoted by $F_0(\omega)$ and $F_1(\omega)$. Decimation with the quincunx matrix will rotate and expand the fan-shaped frequency support, bringing some of its parts outside the $[-\pi, \pi)^2$ frequency cell. Nonetheless, fan-shaped supports provide an alia-free decimation system as explained in [130].

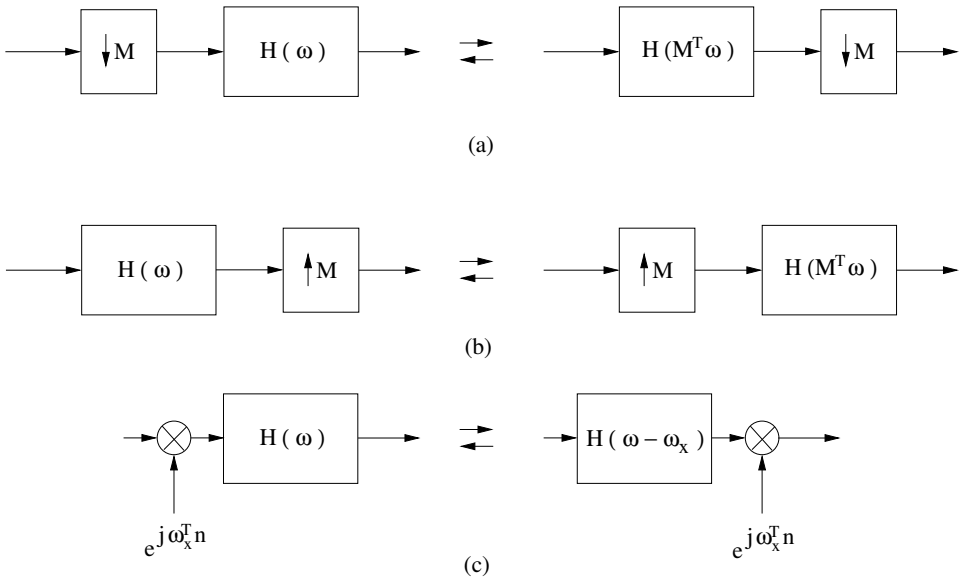


Figure 13: Identities for 2-D multirate systems.

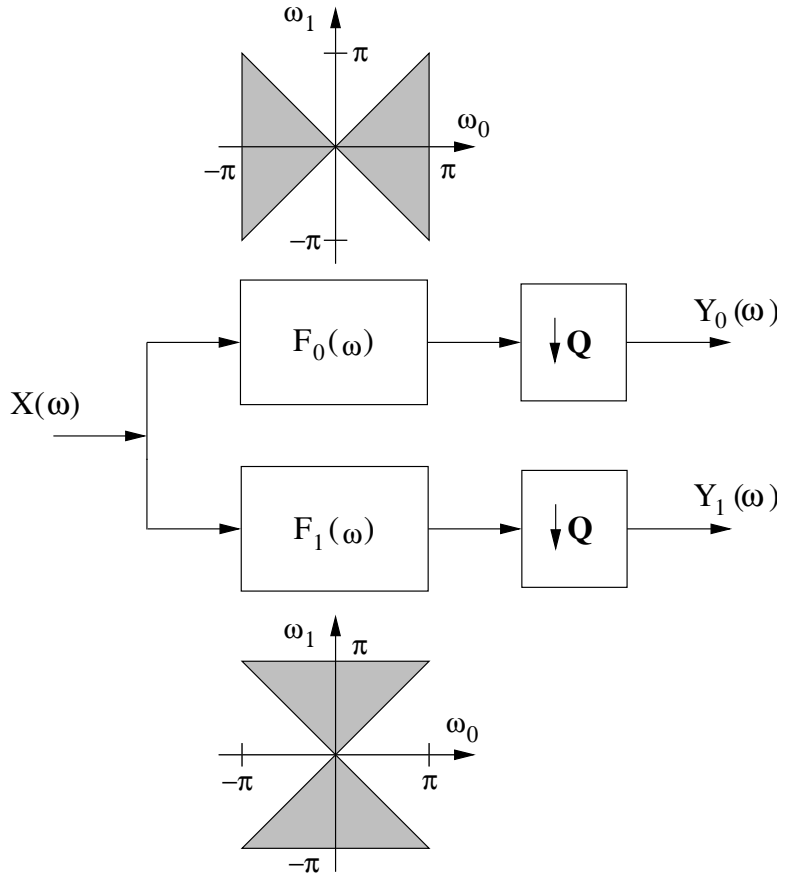


Figure 14: Block diagram for the FFB showing the ideal support for the 2-D filters.

The second issue with the original DFB structure can be understood by looking at the overall downsampling matrix for each channel of the DFB. For instance, an eight-band DFB has an overall downsampling matrix given by

$$\mathbf{D}_\ell = \mathbf{Q}_1\mathbf{Q}_2\mathbf{U}_i\mathbf{Q}_3 \tag{6}$$

for $\ell = 1, 2, ...8$, and $i = \lceil \frac{\ell}{2} \rceil$. This implies that $\mathbf{D}_{2i-1} = \mathbf{D}_{2i}$. The $\mathbf{D}_\ell$ matrix is non-diagonal for a DFB with three or more stages. In turn, this means that each subband lies on a lattice with non-rectangular geometry. Hence, the subband coefficients lie on a grid that makes further intra-subband and inter-subband processing unpractical.

As Bamberger noted in [6], an integer sampling matrix can be expressed as the product of an integer diagonal matrix and a unimodular matrix. Hence, it is possible to rearrange the DFB output on a rectangular lattice by finding the unimodular sampling matrix that diagonalizes $\mathbf{D}_\ell$. Park, et al. [96] have developed a systematic method to obtain these matrices. They have proposed the so-called "backsampling" matrices $\mathbf{B}_\ell$. The objective is to find a matrix that undoes the skewings and rotations induced by the overall resampling matrices $\mathbf{D}_\ell$. Perhaps, a more precise description for backsampling could be "lattice rectangularization".

Here we focus on the eight-band DFB case. As the DFB is maximally decimated, each band must have a decimation ratio of eight, This is easily verified for all $\mathbf{D}_\ell$. There are only two diagonal integer matrices that can provide the desired rectangular geometry. Namely,

$$\mathbf{C}_1 = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \quad \text{and} \quad \mathbf{C}_2 = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}. \tag{7}$$

The problem is then to select the matrix $\mathbf{C}_n$ that satisfies

$$\mathbf{D}_\ell\mathbf{B}_\ell = \mathbf{C}_n, \quad n = 0, 1, \tag{8}$$

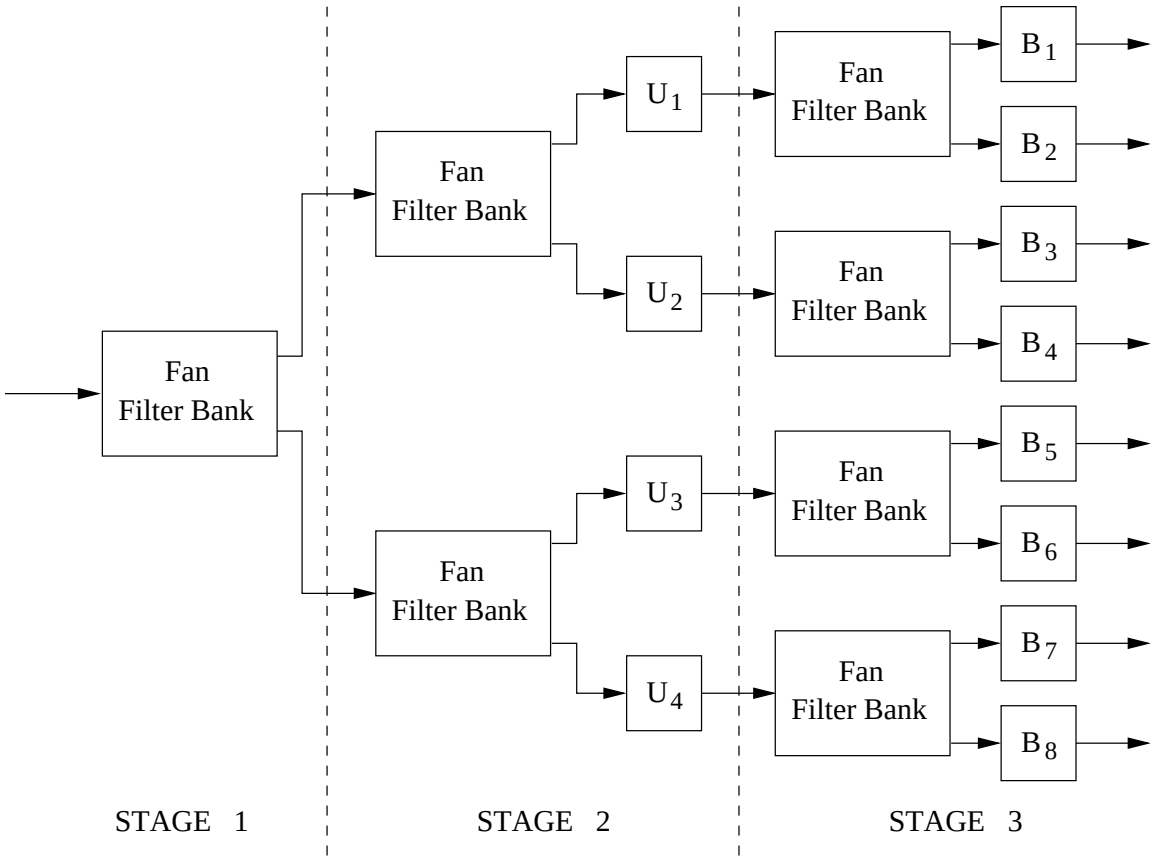


Figure 15: Implementation of an eight-band DFB using a tree structure with FFBs and backsampling matrices.

such that the backsampling matrix $\mathbf{B}_\ell$ has all integer entries and $|\det(\mathbf{B}_\ell)| = 1$. Direct calculation gives the following matrices

$$\mathbf{B}_1 = \mathbf{B}_2 = \begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}, \quad \mathbf{B}_3 = \mathbf{B}_4 = \begin{bmatrix} -1 & -1 \\ 0 & 1 \end{bmatrix},$$
$$\mathbf{B}_5 = \mathbf{B}_6 = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix}, \quad \mathbf{B}_7 = \mathbf{B}_8 = \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}. \tag{9}$$

The updated tree structure is presented in Figure 15. It includes the use of backsampling matrices.

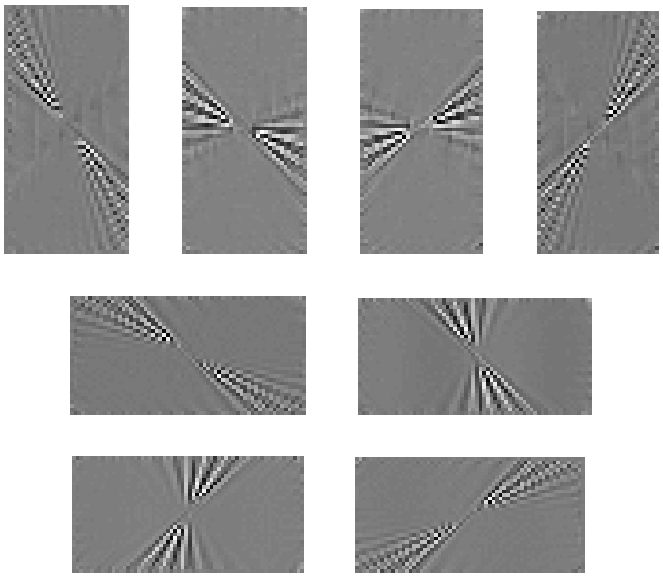


Figure 16: Output of the DFB using fan filter banks and backsampling matrices. The problems of the original DFB have been fixed.

We compare the diamond-based and FFB-based DFBs in Figure 16 with those of Figure 12. The distortion introduced by modulation (checkerboard patterning) and the non-rectangular lattice are present in the diamond filter bank case. The fan-based DFB includes lattice diagonalization using "backsampling" matrices. We clearly see the improvement obtained with the use of FFBs. In the next section we describe the actual design of the diamond and fan filters used in these examples.

We note that subbands $\{1, 2, 3, 4\}$ have an overall diagonal downsampling matrix given by $\mathbf{C}_1$, while subbands $\{5, 6, 7, 8\}$ have $\mathbf{C}_2$. Hence, there is a tradeoff between spatial and angular resolution which depends on the orientation of the subband. For subbands whose axis of symmetry is closer to the ω_0 axis, the vertical frequency resolution decreases as more angular resolution increases. The opposite occurs for subbands closer to the ω_1 axis.

1.2.1 Design of Diamond and Fan Filters

The next step in our discussion of the DFB is to review the design the 2-D diamond-shaped and fan-shaped filters such that the overall analysis-synthesis system meets other desirable

requirements like alias-free reconstruction and/or perfect reconstruction.

Thus far, all the discussion has been based on the availability of filters that meet the ideal passband requirements. In practice, this is obviously not the case. Designing the filters for the diamond and fan filter banks, or other 2-D systems, has been studied by several authors (see discussion in [98]). Different alias-free and perfect reconstruction (PR) 2-D filter banks have been proposed. The filters are in general non-separable and need have large orders to achieve good frequency domain characteristics. Implementation is unattractive from the computational perspective. In particular, Bamberger explored this approach in [5], where he presented a design method for FIR PR systems using McClellan transformations. Polyphase-based separable designs have also been explored, but these tend to destroy the PR property [98]. Another approach that is of interest, is the change of variables scheme devised by Ansari [2] and generalized by Bamberger [5]. First we review this technique for the design of the diamond filter bank [5, 7].

The 2-D filters are obtained from a pair of 1-D prototypes that satisfy the QMF relationship

$$H_0(z) = H_1(-z). \tag{10}$$

It is well known that the polyphase representation for these filters is

$$\begin{aligned} H_0(z) &= E_0(z^2) + z^{-1}E_1(z^2) \\ H_1(z) &= E_0(z^2) - z^{-1}E_1(z^2). \end{aligned} \tag{11}$$

The first step of the design technique consists of first forming a pair of separable checkerboard support filters given by

$$\begin{aligned} C_0(z_0, z_1) &= H_0(z_0)H_0(z_1) + H_0(-z_0)H_0(-z_1) = 2[E_0(z_0^2, z_1^2) + z_0^{-1}z_1^{-1}E_1(z_0^2, z_1^2)] \\ C_1(z_0, z_1) &= H_0(z_0)H_1(z_1) + H_0(-z_0)H_1(-z_1) = 2[E_0(z_0^2, z_1^2) + z_0^{-1}z_1^{-1}E_1(z_0^2, z_1^2)]. \end{aligned} \tag{12}$$

The second crucial step is to apply the following change of variables

$$\begin{aligned} z_0 &\rightarrow z_0^{\frac{1}{2}} z_1^{-\frac{1}{2}} \\ z_1 &\rightarrow z_0^{\frac{1}{2}} z_1^{\frac{1}{2}}. \end{aligned} \tag{13}$$

This step produces the geometric transformation on the checkerboard support to obtain the desired diamond support filters denoted by $H_0(z_0, z_1)$ and $H_1(z_0, z_1)$.

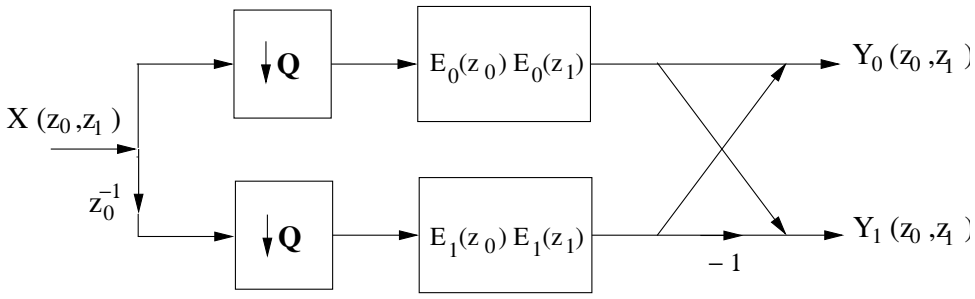


Figure 17: Efficient polyphase domain implementation of diamond filter bank. The filtering operations are separable.

In frequency domain we have

$$\begin{aligned} H_0(\omega) &= E_0(\mathbf{Q}^T\omega) + e^{-j\omega^T\mathbf{k}_1}E_1(\mathbf{Q}^T\omega) \\ H_1(\omega) &= E_0(\mathbf{Q}^T\omega) - e^{-j\omega^T\mathbf{k}_1}E_1(\mathbf{Q}^T\omega) \end{aligned} \tag{14}$$

where $\mathbf{Q}$ is the quincunx matrix, $\mathbf{k}_1 = [10]^T$, and the polyphase components are separable operations given by

$$\begin{aligned} E_0(\omega) &= E_0(\omega_0,\omega_1) = E_0(\omega_0)E_0(\omega_1) \\ E_1(\omega) &= E_1(\omega_0,\omega_1) = E_1(\omega_0)E_1(\omega_1) \end{aligned} \tag{15}$$

The most significant aspect of this implementation resides on the low computational complexity that is obtained by processing in the polyphase domain. As shown in Figure 17, the polyphase filters are formed by separable operations.

For the 1-D prototypes, it is possible to use FIR filters designed by Johnston [59] which are alias free, but not PR. Another choice is to use the IIR linear phase filters proposed by Smith and Eddins in [120]. These filters have the form

$$H_0(z) = \frac{b_0 + b_1z^{-1} + b_2z^{-2} + b_2z^{-3} + b_1z^{-4} + b_0z^{-5}}{a_0 + a_1z^{-2} + a_0z^{-4}}. \tag{16}$$

The IIR filters are non-causal, but can be used for the case of finite length signals (like images), using forward and backward difference equations. Moreover, the implementation is very efficient, requiring only 5.1 multiplies and 5.6 adds per output sample, while achieving similar passband characteristics to that of the 32 tap Johnston filters. The magnitude responses are compared in Figure 18. We show in Figure 19 the magnitude response of the diamond filter $H_0(\omega_0,\omega_1)$ and $H_1(\omega_0,\omega_1)$ obtained from the "32D" Johnston filter tabulated in [59].

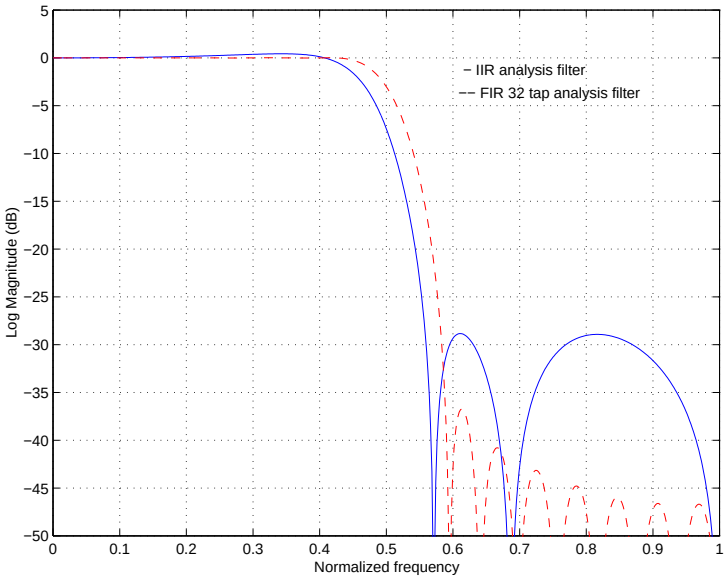


Figure 18: Magnitude response comparison of IIR and FIR 1-D prototypes satisfying $H_1(z) = H_0(-z)$. The IIR filter has significant lower computational complexity.

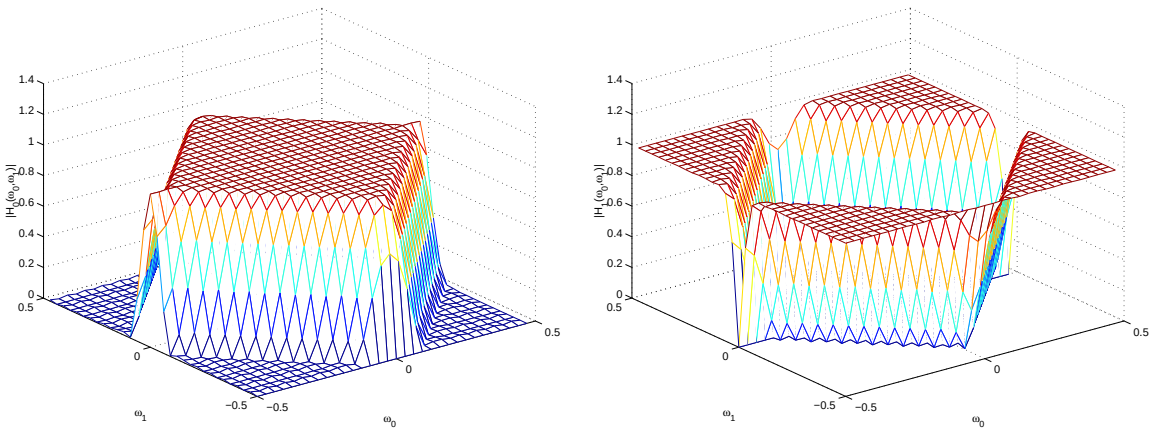


Figure 19: Magnitude of the diamond filter pair synthesized from 32-tap FIR Johnston filters

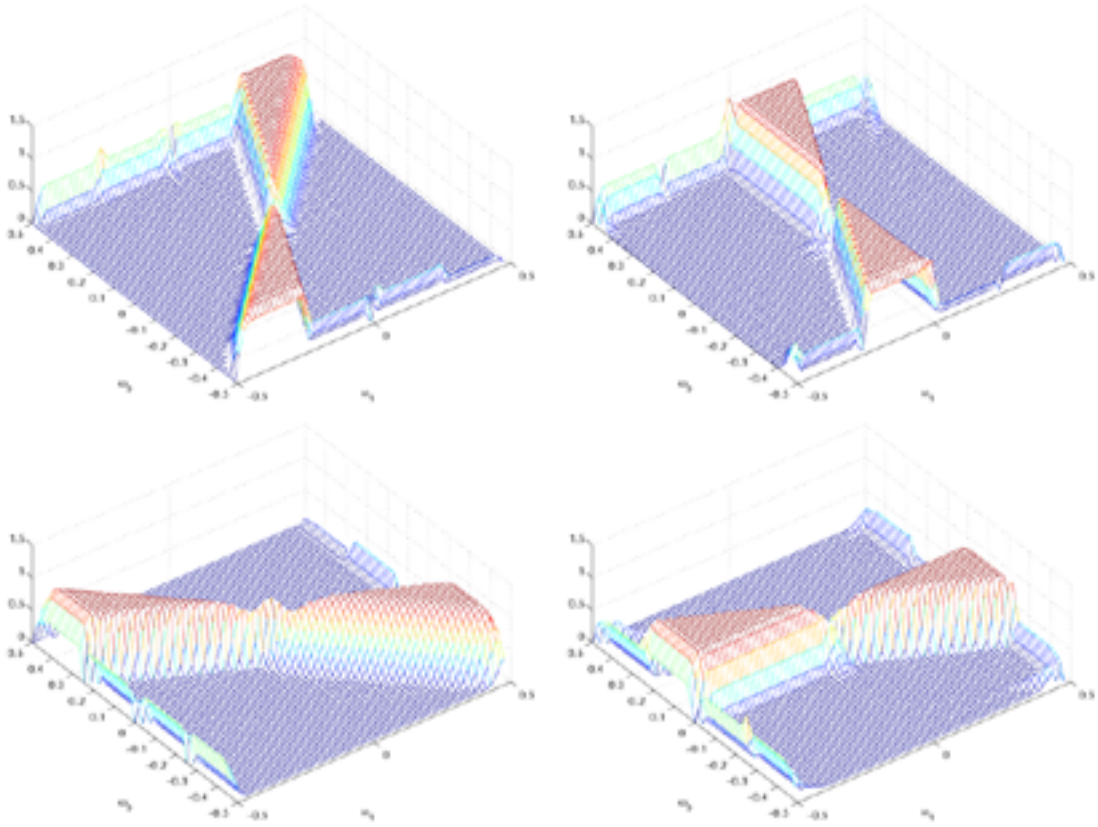


Figure 20: Examples of magnitude responses of the actual wedge filters constructed with the Johnston's 32D FIR filters

This scheme based on the change of variables is quite powerful, and allows for many other possible supports (see [98] for example). For the case of fan filters, we only let $z_0 \rightarrow -z_0$ in the derivation above. Moreover, as will be shown in the next chapter, the condition from Equation 10 is not necessary to achieve this efficient implementation.

Finally, we illustrate in Figure 20 the actual synthesized wedge passband filters for some of the channels on an eight-band DFB using the 32-tap Johnston filters.

1.3 Summary and Remaining Challenges

In this chapter we gave a review of the DFB developed by Bamberger and Smith. Furthermore, we discussed how the original DFB has been improved over the years in order to make it more amenable as a tool for image processing and image analysis.

There are a few issues that remain to be explored:

1. Except for the trivial two tap case, FIR filters of the form $H_0(z) = H_1(-z)$ do not have the PR property. The alternative is to design 2-D nonseparable filters with the desired properties. As discussed, this significantly increases the computational requirements of the DFB. It would be desirable to have a similar efficient polyphase structure that achieves PR using FIR filters.

2. The Bamberger DFB is a maximally decimated structure. Hence, it is missing the shift invariance property. Maximal decimation is useful for compression applications, but shift invariance is highly desirable for other purposes. It remains to find a shift invariant (i.e. undecimated) version of the Bamberger DFB.

3. The DFB is a purely directional representation. Having a radial component would allow us to explore the use of the DFB on applications that exploit the multiresolution information as well.

In the next chapter, we tackle these three issues. This leads to the development of what we call the ladder-based Biorthogonal Directional Filter Bank, to an efficient undecimated DFB structure, and to directional pyramids for image processing and analysis applications.

CHAPTER II

NEW STRUCTURES FOR DIRECTIONAL PROCESSING OF IMAGES

In the previous chapter, we introduced the work of Bamberger and Smith on DFBs. We will refer to this type of DFB as the Bamberger DFB for easy identification. In this chapter, we address some of the open questions regarding this DFB.

First, we introduce the use of ladder structures as an approach to achieve PR with FIR filters while maintaining an efficient computational implementation in the polyphase domain. This implementation arises from a simple 1-D to 2-D transformation of biorthogonal systems implemented as ladder structures. Henceforth, this system will be called a ladder-based biorthogonal DFB.

Second, we introduce a novel undecimated directional filter bank. This structure is obtained by manipulating the resampling matrices in the Bamberger DFB, such that low computational complexity is preserved, while achieving shift invariance that could be useful in pattern recognition and image denoising applications.

Finally, as the Bamberger DFB is a purely directional image decomposition, it lacks the ability to distinguish details based on low, medium and high frequencies. In this chapter we introduce pyramidal structures that provide low-to-high frequency analysis in combination with the DFB. We report different structures with different degrees of redundancy. As will be seen in subsequent chapters, these directional pyramids provide excellent results in image analysis and enhancement applications.

2.1 Implementation of two-channel filter banks using ladder structures.

First we give a brief review of ladder structures, the basis for implementing the biorthogonal DFB. We should note that in the wavelet community, ladder structures have been

introduced as the lifting scheme. These structures have also been introduced as Perfect Inversion Networks (PINs) in [13] as a means of eliminating quantization noise. Henceforth we consider ladder structures, lifting, and PINs as equivalent concepts, at least as far as processing discrete time signals. We use ladder terminology as these structures have been known in the filter bank community since the early seventies [116, 88, 24].

A biorthogonal two-channel filter bank consists of analysis filters $h_0[n]$, $h_1[n]$ and synthesis filters $g_0[n]$, $g_1[n]$ that satisfy the conditions

$$H_0(z^{-1})G_0(z) + G_1(z)H_1(z^{-1}) = 1 \tag{17}$$

$$H_0(-z^{-1})G_0(z) + G_1(z)H_1(-z^{-1}) = 0, \tag{18}$$

where $H_0(z)$, $H_1(z)$, $G_0(z)$, and $G_1(z)$ are the z-transforms of the filter set, and

$$H_1(z) = z^{-1}G_0(-z^{-1}) \qquad H_0(z) = -z^{-1}G_1(-z^{-1}). \tag{19}$$

These equations establish the PR and alias cancellation conditions for the system. In practice a delay is added to the filters to have a causal implementation.

A useful analysis tool in filter bank theory is the polyphase representation of the filter bank shown in Figure 21-(a), which allows arithmetic to be performed at the lower sampling rate as depicted in Figure 21-(b). $\mathbf{E}(z)$ and $\mathbf{R}(z)$ are the well known analysis and synthesis polyphase matrices. The entries of the matrices consist of polynomials in z given by

$$E_{i,j}(z) = \sum_n h_i(2n + j)z^{-n}$$

$$R_{i,j}(z) = \sum_n g_i(2n + 1 - j)z^{-n}$$

for $i, j = 0, 1$. The PR condition implies that

$$\mathbf{R}(z)\mathbf{E}(z) = \mathbf{I}. \tag{20}$$

Since $\mathbf{R}(z) = \mathbf{E}(z)^{-1}$, this implies that $\det(\mathbf{R}(z)) = \det(\mathbf{E}(z)) = 1$.

We next introduce the concept of a ladder structure. Suppose we have the following unimodular matrix, and its corresponding inverse

$$\mathbf{U}(z) = \begin{bmatrix} 1 & u(z) \\ 0 & 1 \end{bmatrix} \quad \mathbf{U}^{-1}(z) = \begin{bmatrix} 1 & -u(z) \\ 0 & 1 \end{bmatrix} \tag{21}$$

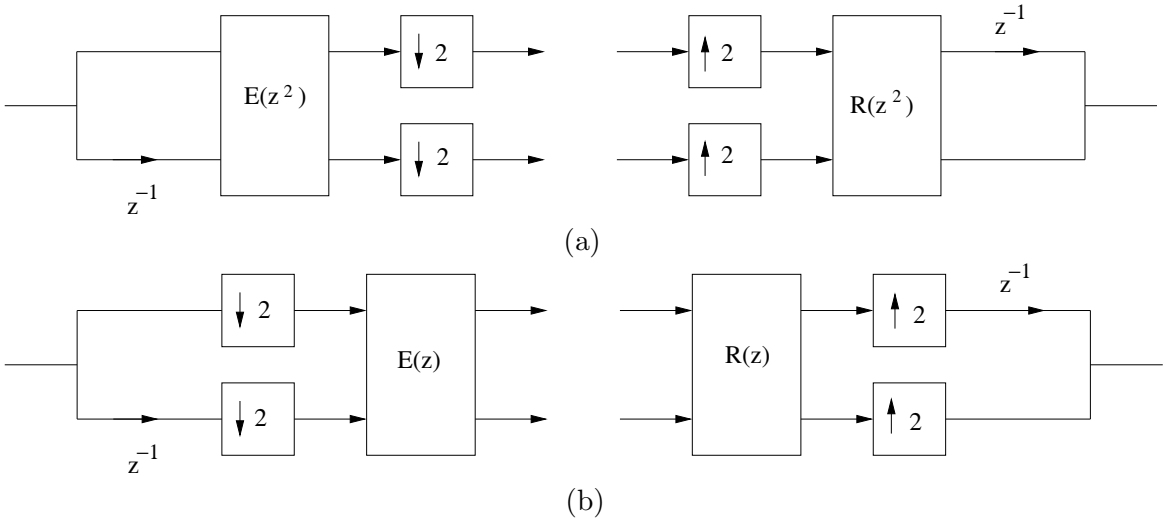


Figure 21: Polyphase representation of two-channel filter bank. (a)In full-rate domain (b)In downsampled domain.

where $u(z)$ is a finite length polynomial. We can modify the biorthogonal system represented by 20 without losing the PR property by letting $\mathbf{R}_U(z) = \mathbf{R}(z)\mathbf{U}^{-1}(z)$ and $\mathbf{E}_U(z) = \mathbf{E}(z)\mathbf{U}(z)$. The new polyphase matrix product is

$$\mathbf{R}_U(z)\mathbf{E}_U(z) = \mathbf{I}. \tag{22}$$

Clearly the determinant remains one in both cases, and PR is preserved. Graphically, the modified polyphase-domain structure of a two channel system is modified as in Figure 22-(a). The modification of the standard two-channel structure is shown in Figure 22-(b).

Lower triangular unimodular matrices show similar properties. For instance, suppose we have

$$\mathbf{L}(z) = \begin{bmatrix} 1 & 0 \\ \ell(z) & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{L}^{-1}(z) = \begin{bmatrix} 1 & -\ell(z) \\ 0 & 1 \end{bmatrix}. \tag{23}$$

We can also form a new set of polyphase matrices given by

$$\mathbf{R}_U(z)L^{-1}(z)L(z)\mathbf{E}_U(z) = \mathbf{I}. \tag{24}$$

In this case, we see that the original system now assumes the modified form shown in Figure 23, which is called a ladder network/stucture. We refer to the unimodular matrices

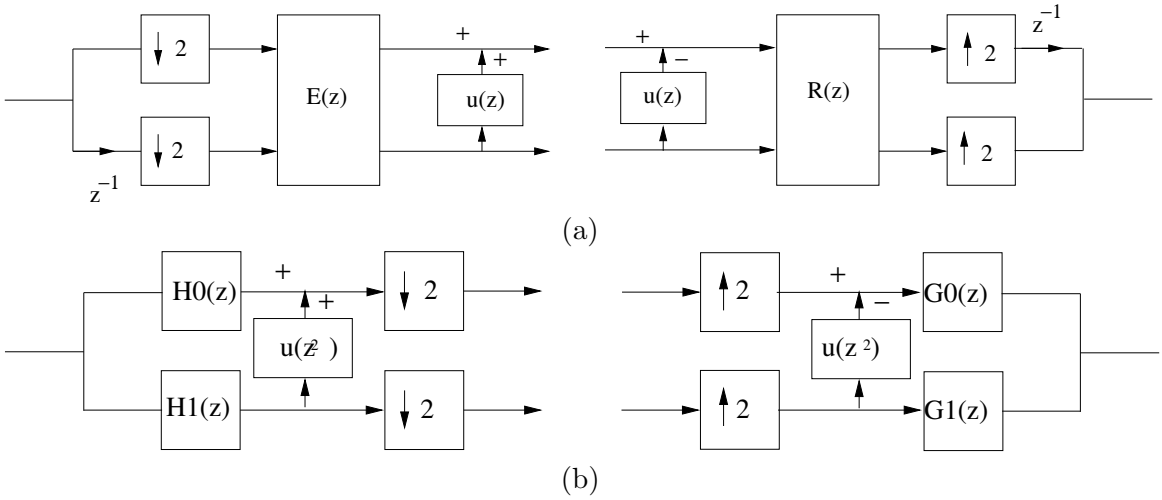


Figure 22: Graphical illustration on a two-channel filter bank modification by an unimodular upper triangular matrix. (a)Modification of polyphase structure. (b)Modification of two-channel structure.

of Equations (21) and (23) as ladder-step matrices or ladder-steps for short. The PR property is guaranteed structurally by sequentially adding the forward and reverse steps to the analysis and synthesis stages in the proper order. The remaining step is to determine the ladder-step filters so that the resulting filters $H_0'(z), H_1'(z), G_0'(z), G_1'(z)$ have good lowpass and highpass characteristics.

Consider the following example. Start with the simple delay chain where $H_0(z) = 1$ and $H_1(z) = z^{-1}$. The polyphase matrices are simply $\mathbf{E}(z) = \mathbf{R}(z) = \mathbf{I}$. Assuming $u(z^2)$ and $\ell(z^2)$ are two zero-phase finite polynomials, we can build a new biorthogonal filter bank using lower and upper triangular ladder steps

$$\mathbf{E}(z) = \begin{bmatrix} 1 & 0 \\ \ell(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & u(z) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & u(z) \\ \ell(z) & \ell(z)u(z)+1 \end{bmatrix} \tag{25}$$

$$\mathbf{R}(z) = \begin{bmatrix} 1 & -u(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\ell(z) & 1 \end{bmatrix} = \begin{bmatrix} 1+\ell(z)u(z) & -u(z) \\ -\ell(z) & 1 \end{bmatrix}. \tag{26}$$

The corresponding analysis filters are then given by

$$H_0(z) = 1 + z^{-1}u(z^2) \tag{27}$$

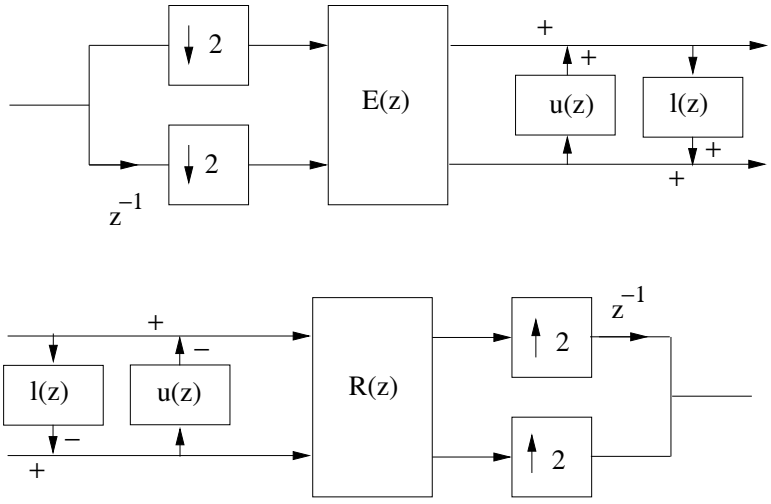


Figure 23: Graphical illustration on a two-channel filter bank modification by upper and lower ladder steps

$$H_1(z) = z^{-1} + \ell(z^2)H_0(z). \tag{28}$$

The synthesis filters are obtained from Equation (19). It is easy to check the PR conditions for this example. More algebraically complex expressions can be obtained for systems with more ladder steps.

From the previous example, it is seen that we can design many types of biorthogonal systems or conversely, factor existing biorthogonal systems into ladder structures. The following two-part theorem by Kalker and Shah [65] summarizes this concept.

Theorem 1 *1. A 2-channel filter bank constructed from a ladder structure constitutes a biorthogonal filter bank.*

2. Every 2-D 2-channel biorthogonal filter bank can be realized with a ladder structure.

Point two of Theorem 1 indicates that all biorthogonal filter sets can be factored into a sequence of ladder steps. Factorization of two channel filter banks into a ladder structure has been reported by different researchers [65, 122, 27]. The factorization is based on a well known result: 2×2 matrices of polynomials with determinant equal to one can be factored into a series of elementary and unimodular matrices.

A well known example of factorization, which has become part of the JPEG2000 standard, is the ladder structure for the 9-7 filters [27] . The factorization of the analysis polyphase matrix is given by

$$\mathbf{E}(z) = \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix} \begin{bmatrix} 1 & \delta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma(1+z) & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & \beta(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \alpha(1+z) \\ 0 & 1 \end{bmatrix}, \tag{29}$$

where $\alpha = -1.586134$, $\beta = -0.05298$, $\gamma = 0.882911$, $\delta = 0.443506$, and $\zeta = 1.149604$.

The construction approach for biorthogonal systems using ladder networks has been reported in various works. In this thesis we follow the approach of Phoong et al. [98] and Ansari et al. [3], which provides straightforward 1-D to 2D mappings. In [98], a family of biorthogonal two-stage ladder filters are designed, which is essentially the same system as the one described by Equations (25) and (26). Their particular realization of this filter bank is given by

$$\mathbf{E}(z) = \begin{bmatrix} 0.5 & 0 \\ -0.5\beta(z) & 1 \end{bmatrix} \begin{bmatrix} z^{-N} & \beta(z) \\ 0 & z^{-2N+1} \end{bmatrix} \tag{30}$$

$$\mathbf{R}(z) = \begin{bmatrix} z^{-2N+1} & -\beta(z) \\ 0 & z^{-N} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.5\beta(z) & 0.5 \end{bmatrix} \tag{31}$$

from which we obtain the analysis filters

$$H_0(z) = \frac{(z^{-2N} + z^{-1}\beta(z^2))}{2} \tag{32}$$

$$H_1(z) = -\beta(z^2)H_0(z) + z^{-4N+1}. \tag{33}$$

In contrast to our previous example, the delays to generate causal filter responses have been added to $\mathbf{E}(z)$ and $\mathbf{R}(z)$. Second and more important, is that they have chosen for $u(z)$ and $\ell(z)$ the same filter response $\beta(z)$. Hence, the design and quality of the filters depends on a single function. The corresponding ladder structure is presented in Figure 24.

Equation (32), is the expression of a halfband filter. To make $H_0(z)$ an ideal lowpass

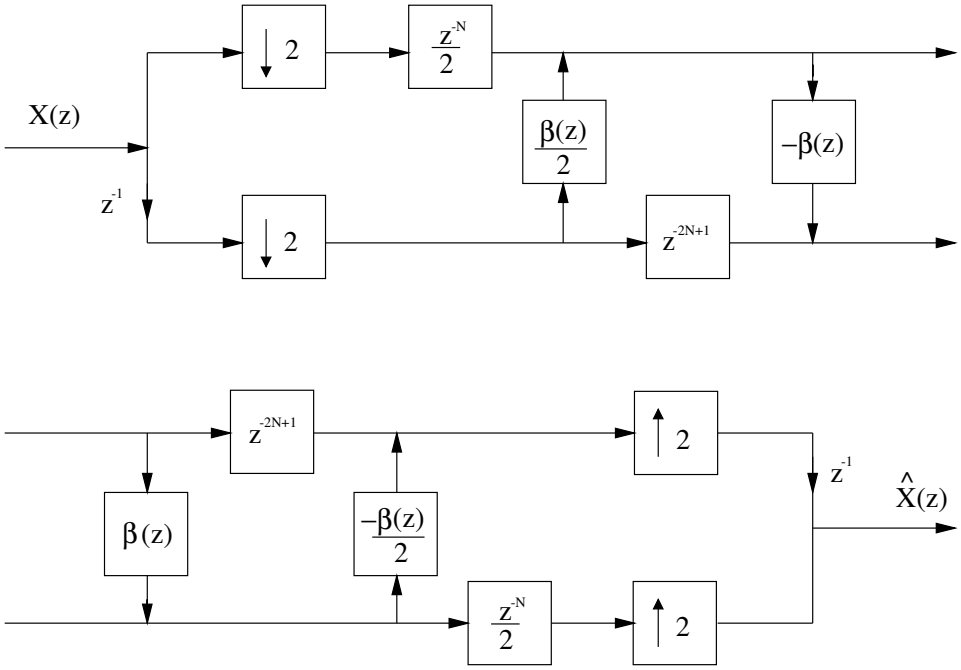


Figure 24: Analysis and synthesis ladder structures for a system with two ladder steps as proposed in [98]

filter and $H_1(z)$ an ideal highpass filter, the following condition must be met

$$\beta(e^{j2\omega}) = \begin{cases} e^{j(-2N+1)\omega}, & \text{for} \quad 0 \leq \omega \leq \pi/2 \\ -e^{j(-2N+1)\omega}, & \text{for} \quad \pi/2 < \omega \leq \pi \end{cases} \tag{34}$$

which implies $\beta(e^{j\omega})$ has allpass behavior.

As proposed in [98], an FIR solution that approximates Equation (34) is obtained by designing an even length, linear phase function with a magnitude response optimized to approximate unity. This is a very simple requirement that can be satisfied with widely available filter design algorithms. We note that $H_0(z)$ and $H_1(z)$ inherit the linear phase property of $\beta(z)$.

In Figure 25, we show the filters obtained by designing $\beta(z)$ with the `fir1` window design routine from MATLAB. The examples include the Hamming window, Bartlett window, Kaiser window, and Chebyshev window. In all cases, the order of $\beta(z)$ was 11 ($L = 12$), and the cutoff frequency was set to $\omega_p = 0.99\pi$.

Two design methods are described in [98]. First, $\beta(z)$ is designed using the Parks-McClellan algorithm by setting the passband edge to 0.8π and the stopband edge to π. The length of $\beta(z)$ was set to 12. The analysis filters are shown in Figure 26. For the second, maximally flat designs are considered using the closed-form formula

$$v_k = \frac{(-1)^{N-k-1} \prod_{i=0}^{2N}(N+1/2-i)}{2(N-k)!(N-1+k)!(2k-1)} \tag{35}$$

where N is the half length of $\beta(z)$ and $\beta_{N-k} = \beta_{N+k-1} = v_k$ for $k = 1, 2, ..., N$. An example of maximally-flat filter design is shown in Figure 27. The maximally flat design method is relevant for the generation of regular biorthogonal wavelets. In this case, Equation (35) guaranties that $H_0(z)$ is constrained to have $2N$ zeros at π, and $G_0(z)$ will have at least $2N$ zeros at π.

All filter sets designed have fairly good frequency responses, except for the presence of overshoot in the $\omega = \pi/2$ vicinity for $H_1(e^{jw})$. As explained in [3], this overshoot is inherent to the two-stage ladder system and cannot be removed by increasing the length of $\beta(z)$.

To improve the frequency response of the two-stage ladder structure, it is necessary to increase the number of ladder stages. This provides the ability to have better control of the passbands of $H_0(z)$ and $G_0(z)$. An extension to three ladder steps has been reported by Ansari, et al. in [3]. The analysis polyphase matrix is given by

$$\mathbf{E}(z) = \begin{bmatrix} 1 & 0 \\ -p_2\beta_2(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & zp_1\beta_1(z) \\ 0 & 1/(1+p) \end{bmatrix} \begin{bmatrix} p_0 & 0 \\ -p\beta_0(z) & 1 \end{bmatrix} \tag{36}$$

which generates the filters

$$H_0(z) = p_0 + p_1 z\beta_1(z^2)A(-z) \tag{37}$$

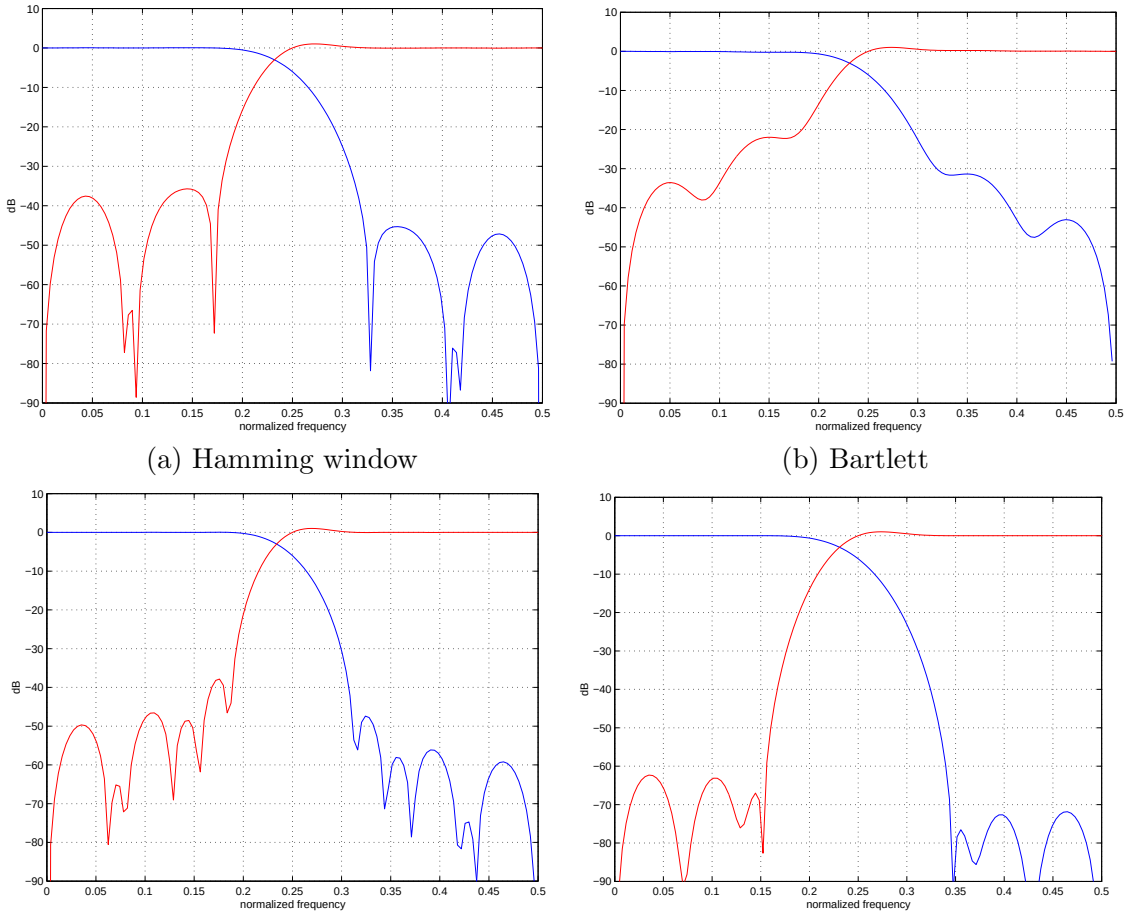
$$H_1(z) = A(-z) - p_2 H_0(z)\beta_2(z^2), \tag{38}$$

where

$$A(z) = \frac{(1 + z\beta_0(z^2))}{1+p} \tag{39}$$

and the constants p_0, p_1, p_2 are given by

$$p_0 = p_1 = \frac{1+p}{2}$$



(a) Hamming window

(b) Bartlett

(c) Kaiser window

(d) Chebyshev window

Figure 25: Examples of magnitude responses generated by the two stage ladder filter bank with $\beta(z)$ designed with the window algorithm. Note the overshoot of the high pass response around $\pi/2$.

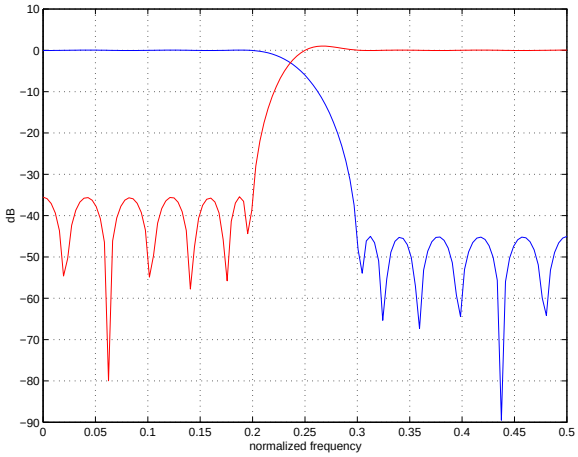


Figure 26: Examples of magnitude responses generated by the two-stage ladder filter bank with $\beta(z)$ designed with the Parks-McClellan algorithm.

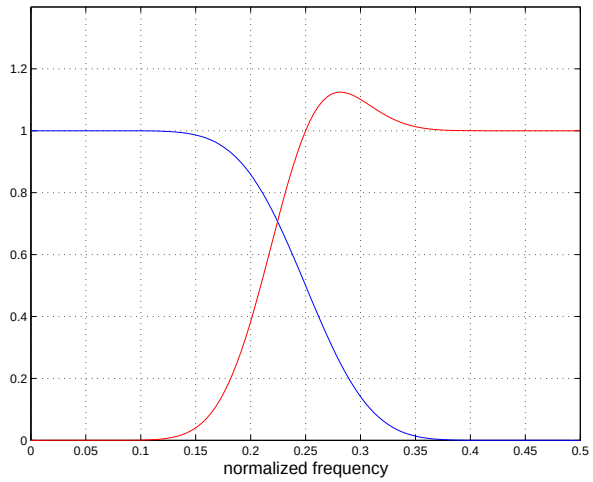


Figure 27: Examples of magnitude responses generated by the two stage ladder filter bank with $\beta(z)$ designed with the maximally-flat algorithm.

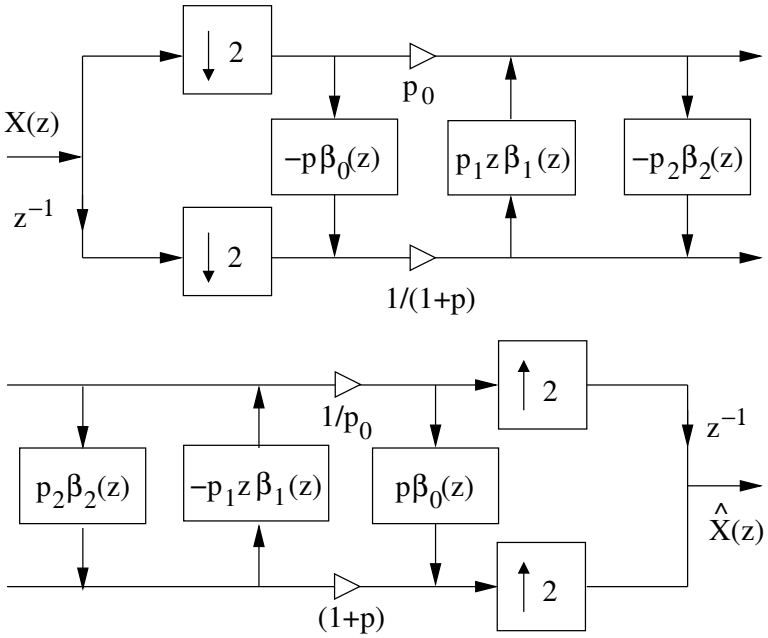


Figure 28: Analysis ladder structure for a system with three ladder-steps as proposed in [3]

$$p_2 = \frac{1-p}{1+p}.$$

These constants are used to control the filter responses of $H_0(z)$ and $H_1(z)$. For instance, p can be selected such that $p_0 = 1/\sqrt{(2)}$. This in turn sets $H_0(e^{j\frac{\pi}{2}}) = H_1(e^{j\frac{\pi}{2}}) = p_0$. As a result, the overshoot at $\pi/2$ for the two-stage ladder is eliminated. We present in Figure 28 the three step ladder structure.

It should be noted that the filters $\beta_0(z)$, $\beta_1(z)$, and $\beta_2(z)$ in Equation 36 are not necessarily the same. Design methods to improve the passbands of the filters are developed in [3]. Each transfer function $\beta_i(z)$ can be used to control different aspects of the responses using the Remez design program. However, we found that for larger step filter lengths, i.e., $L > 6$, setting $\beta_0(z) = \beta_1(z) = \beta_2(z) = \beta(z)$ performs as well as the independent design of $\beta_i(z)$. As an example, we present filters obtained by designing $\beta(z)$ with $L = 8$ using the Parks-McClellan algorithm. An example of the frequency response is shown in Figure 29. This system has similar computational complexity to that of the two stage case, but as can be seen, the passbands don't have overshoot.

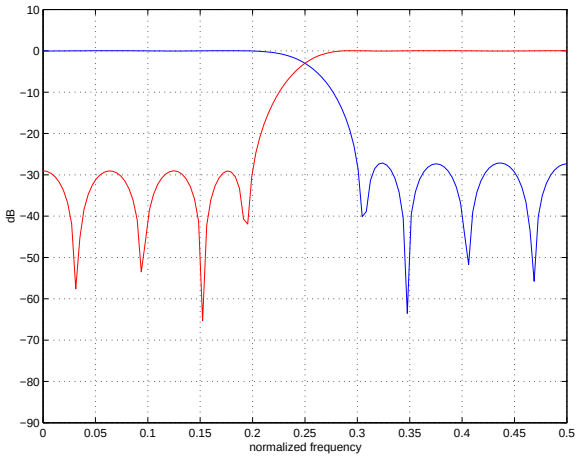


Figure 29: Examples of magnitude responses generated by the three stage ladder filter bank with $\beta(z)$ designed with the Parks-McClellan algorithm. Note that the bump at $\pi/2$ has been eliminated

2.2 Implementation of 2-D ladder based diamond and fan filter banks

For our purposes, the most important feature of ladder structures is that it allows implementation of 2-D diamond and fan filter banks with very simple 1-D to 2-D transformations.

A contribution of this thesis is to marry the biorthogonal ladder filter banks with the Bamberger directional filter bank. Through this union, the Bamberger DFB inherits the features and characteristics of ladder structures. These features are:

1. Implementation of biorthogonal FIR DFBs. Previously, only a non-causal IIR solution allowed PR with linear phase.

2. Efficient separable implementation in the polyphase domain. Since the ladder filters are symmetric, the number of multiplies is half the length of the ladder filters.

3. 2-D filter quality can be controlled by a single function $\beta(z)$.

4. For applications where filter regularity is needed, $\beta(z)$ can be designed so that the analysis and synthesis lowpass filters meet specified regularity constraints.

5. The PR property is preserved structurally, which allows for fixed point implementation of filters without losing filter quality.

6. It is possible to add non-linear processing elements to the ladder stages. As an example, a DFB over finite fields, has been reported by Randolph and Smith in [108] with applications to binary image analysis.

The 1-D to 2-D transformation reported in [98] works directly on the polyphase matrices $\mathbf{E}(z)$ and $\mathbf{R}(z)$ once they are written as a series of ladder steps. This particular transformation is suited for sampling matrices with determinants equal to two, as in the quincunx matrix case.The biorthogonal filter banks of Figures 24 and 28 can be converted to a diamond filter bank by performing the following simple 1-D to 2-D mapping with the entries of $\mathbf{E}(z)$ and $\mathbf{R}(z)$:

1. Replace the 1-D transfer function $\beta(z)$ with the separable 2-D transfer function $\beta(z_0)\beta(z_1)$.

2. Replace the 1-D delays z^{-1} with the 2-D delay $z_0^{-1}z_1^{-1}$.

Using the proper multirate identities from Figure 13, the analysis and synthesis filters for the two-stage 2-D ladder can be expressed as

$$\begin{aligned} H_0(z_0, z_1) &= \frac{1}{2}(z_0^{-2N} + z_0^{-1}\beta(z_0 z_1^{-1})\beta(z_0 z_1)) \\ H_1(z_0, z_1) &= -\beta(z_0 z_1^{-1})\beta(z_0 z_1)H_0(z_0, z_1) + z_0^{-4N+1} \\ G_0(z_0, z_1) &= -H_1(-z_0, -z_1) \\ G_1(z_0, z_1) &= H_0(-z_0, -z_1). \end{aligned} \tag{40}$$

The expressions for the three-stage ladder system are easily derived from Equation (36).

By introducing a frequency shift by π in the ω_0 dimension, or equivalently letting $z_0 \rightarrow -z_0$ in Equation (40), filters with fan-shaped support are generated. Actual fan filters obtained with this transformation are presented in Figure 30 corresponding to the 1-D filters of Figure 29. From the 2-D two-channel filter bank depicted in Figure 31, we see that the ladder steps consist of separable filters.

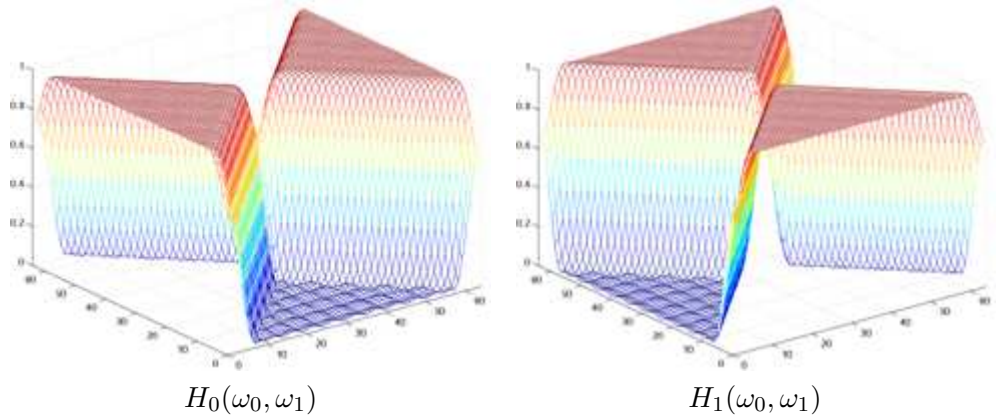


Figure 30: Magnitude response of the analysis fan filters synthesized with the ladder structure.

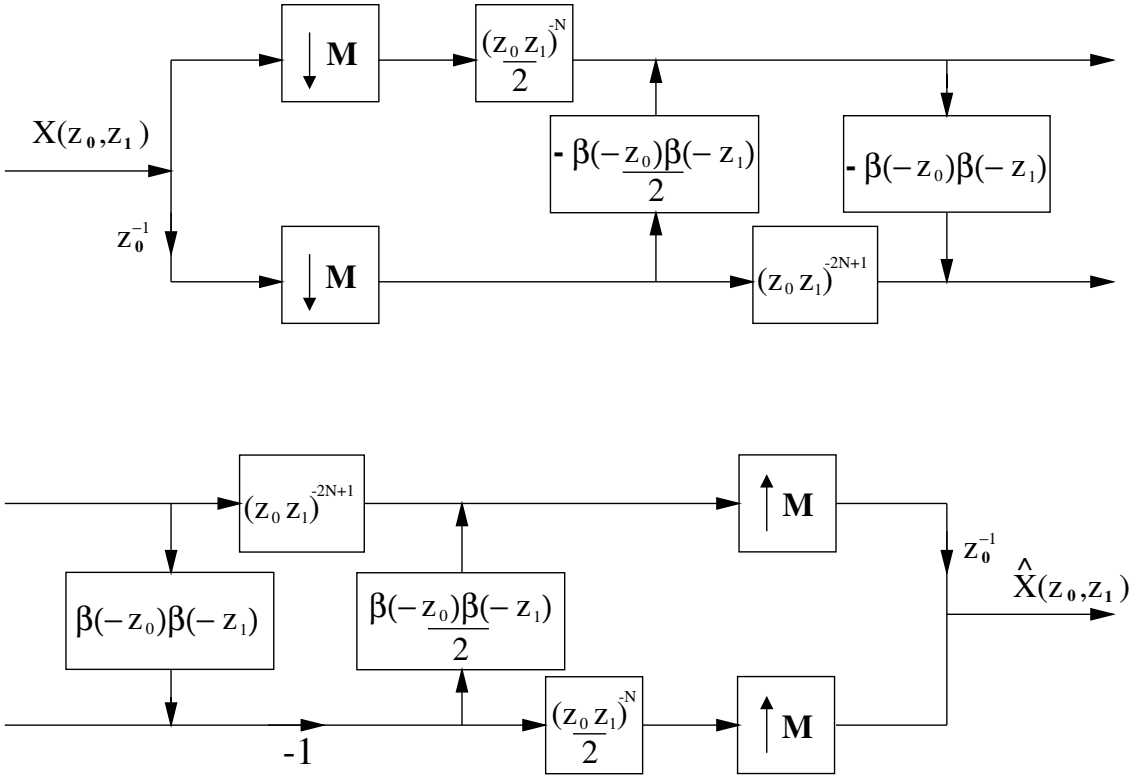


Figure 31: Ladder structure for the implementation of a 2-D two-channel biorthogonal filter bank.

2.3 The Ladder-Based Biorthogonal Directional Filter Bank

From the developments in the previous section, it becomes straightforward to obtain a new type of tree-structured DFB by simply replacing the original QMF structure from Figure 17 with the analysis and synthesis structures of Figure 31. The tree-structure of the DFB, including downsampling, unimodular and backsampling matrices remain unchanged. Since the main distinction of this DFB is that it derives from 1-D biorthogonal ladder-based systems, we call this DFB the biorthogonal ladder directional filter bank to differentiate it from other implementations. Moreover, the biorthogonal property is preserved by the 1-D to 2-D mapping.

In Figure 32 we show the actual magnitude and impulse responses of the analysis filters for a tree structured eight-band DFB using the 1-D $\beta(z)$ function designed with the Parks-McClellan algorithm in the previous section. The filter bank uses a three-stage ladder structure.

2.4 Low-Complexity Undecimated Directional Filter Banks

As with other maximally-decimated representations, the Bamberger DFB is not a shift-invariant representation. In some applications, shift invariance is considered an important property. For the case of the 2-D DWT, shift-invariance is achieved by removing the down-sampling operations from the filter bank structure. This is the so-called undecimated DWT. Unser [128], among others, shows that this decomposition is a wavelet frame and PR is preserved. While the undecimated representation is shift invariant, the coefficient decorrelation is lost.

For similar reasons, it might be desirable to have an undecimated Bamberger DFB. However, the efficient polyphase structures presented thus far rely on working in the down-sampled polyphase domain. Two approaches have been proposed to obtain undecimated DFBs. However, as described next, they do not keep all the properties of the DFB.

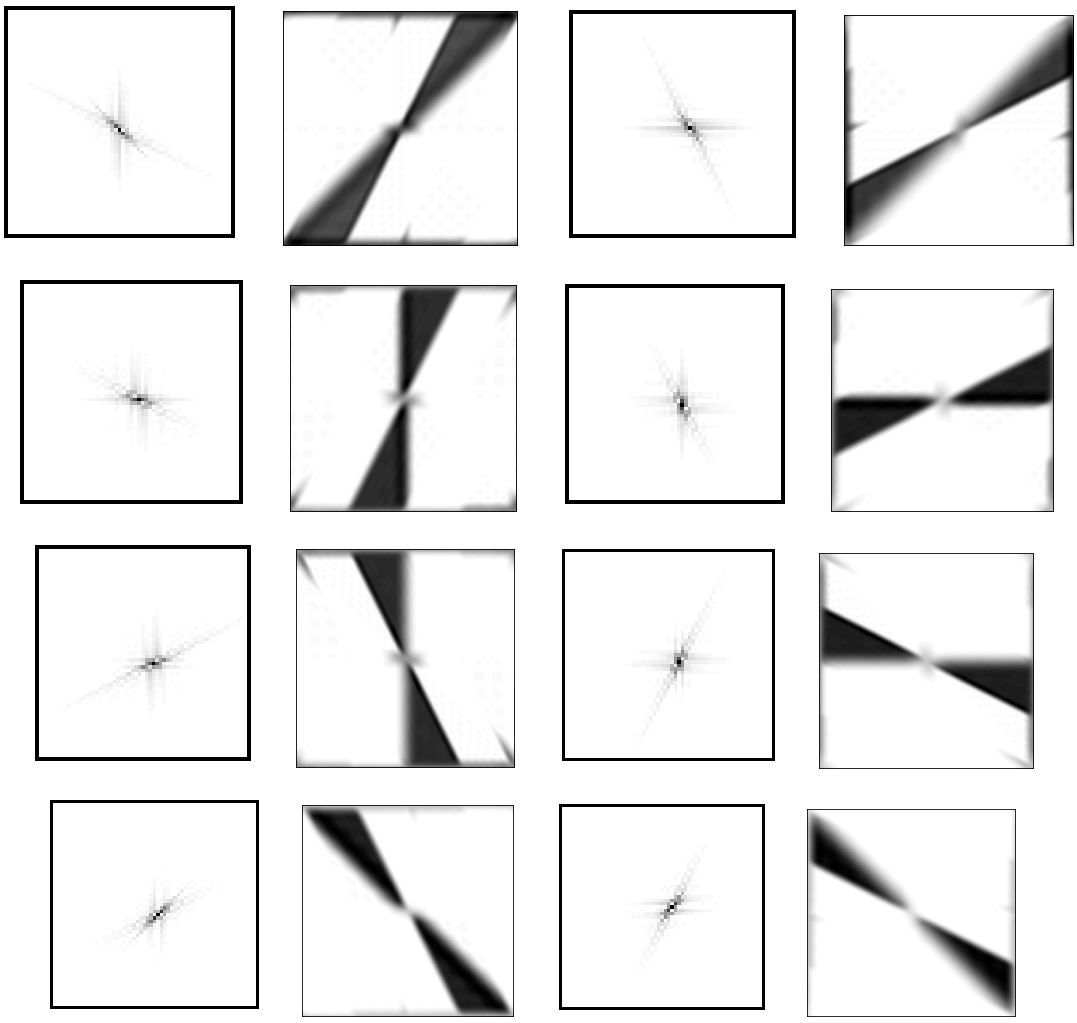


Figure 32: Impulse and magnitude responses of the synthesis filters of an 8 band biorthogonal ladder DFB

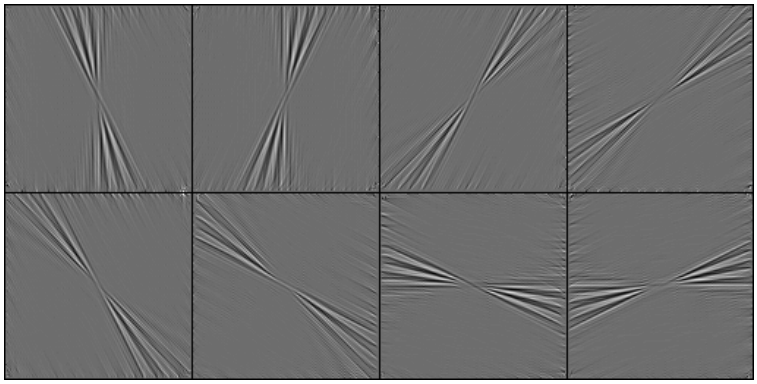
2.4.1 Previous work

Bamberger [5], proposed to give up the efficient structure and design non-separable 2-D filters (diamond and parallelogram) that satisfy the PR conditions. Then an undecimated DFB would be obtained by removing all the resampling matrices from the analysis and synthesis structures. However, as expected, the computational complexity of non-separable filters makes this approach rather unattractive, since the filter dimensions need to be large in order to realize good magnitude responses.

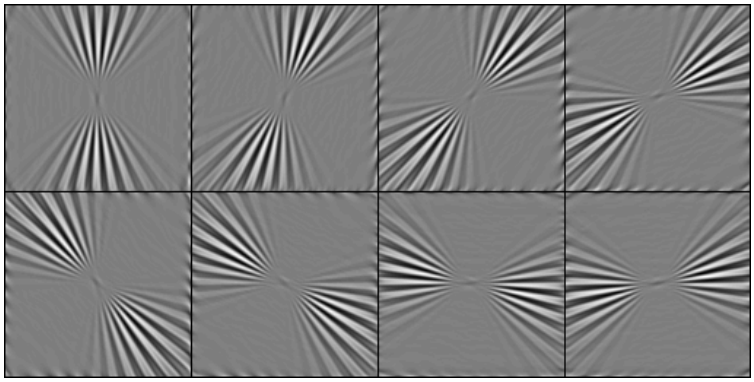
The second approach used in [95], is what we would identify as a back-end full-rate (BEFR) DFB. This representation consists of computing an N-band DFB decomposition and then generating a full-rate band by passing each subband separately through the synthesis stage. Unfortunately, this is not an undecimated transform since the subbands have been through a decimation-interpolation process. Hence the subbands will have some amount of aliasing content which is cancelled when the subbands are added together. The effects of aliasing should be minimal and unnoticeable when the filters have good magnitude responses. Although the BEFR DFB has a significant increase in complexity with respect to the DFB, it can have much lower complexity than the non-separable 2-D implementation by taking advantage of the symmetries of ladder structures and parallelism of the DFB tree structure.

As an example, we show the resulting eight-band BEFR DFB decomposition for the "vertigo" test image from MATLAB in Figure 33-(a), using a $\beta(z)$ with $L = 16$, and Parks-McClellan design. For comparison purposes, we show the corresponding eight-band decomposition for the finest level of the steerable pyramid from Simoncelli, et al. [118] in Figure 33-(b). The directional selectivity of the DFB is noticeably better.

The steerable pyramid presents low quality on resolving a given direction; this is expected since the steerable pyramid uses functions of raised cosines to build the magnitude responses of the filters. Hence, there is little control on the passband and stopband characteristics of the filters, allowing for energy leakage into adjacent directions and towards the lowpass region.



(a) Eight band Back-end full-rate DFB decompsition



(b) Eight band steerable pyramid decomposition (finest scale)

Figure 33: Comparison of quality of directional information for the DFB (back-end full-rate) vs. the steerable pyramid

2.4.2 Low-Complexity Fan Filter Banks

In this subsection we introduce a manipulation of the Bamberger DFB structure that allows us to generate a truly undecimated directional decomposition which remains as computationally efficient as the maximally decimated DFB. By efficient we mean that the filtering operations remain separable, while the passbands are realized without introducing aliasing from the downsampling operations.

We start by considering the two-channel fan filter bank structure. This structure serves as the building block for the tree-structured DFB. We depict the general polyphase structure in Figure 36-(a), where $\mathbf{M}$ is a quincunx resampling matrix. For this discussion assume

$$\mathbf{M} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}.$$

As we have discussed, $\mathbf{M}$ has a downsampling factor of two and inflicts a 45 degree rotation on the data. This rotation is what allows for the separability of the filtering operations. Conversely, upsampling by $\mathbf{M}$ introduces a rotation opposite in direction and inserts zeros in the coset lattice.

An undecimated decomposition is easily obtained from the maximally decimated structure. First, we move the polyphase matrices from the lower rate to the higher rate using the multirate identities (Figure 34-(a),(b)) as shown in Figure 35. Second, we drop the resampling matrices. This approach is similar to the one suggested by Bamberger, but has the advantage that the filters are still derived from 1-D prototypes. On the other hand, the filters are also non-separable and computational efficiency is lost.

For this reason, it becomes desirable to find an implementation of the DFB that retains the order of complexity from the Bamberger DFB, or more precisely, scales linearly with respect to the number of samples. We identify the following requirements:

1. No data can be discarded.

2. We need to retain separability of the filtering operations.

This in turn suggests that the downsampling and rotation properties of $\mathbf{M}$ need to be decoupled somehow.

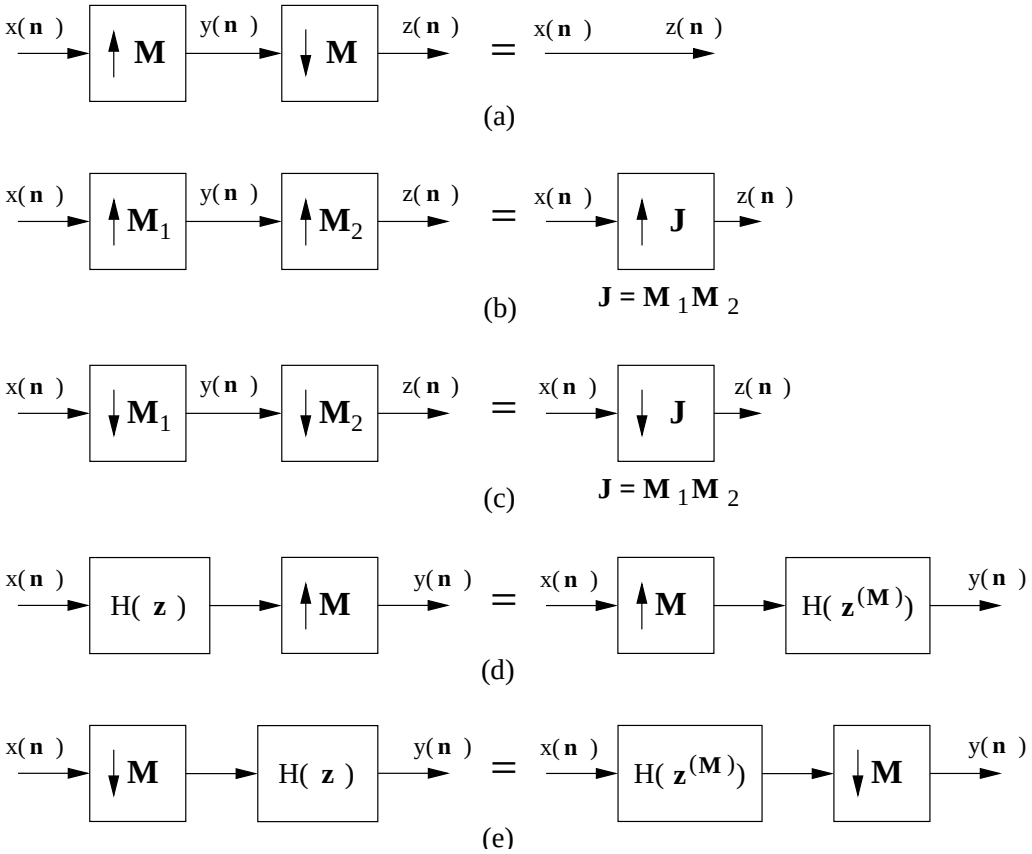


Figure 34: Useful multirate identities.

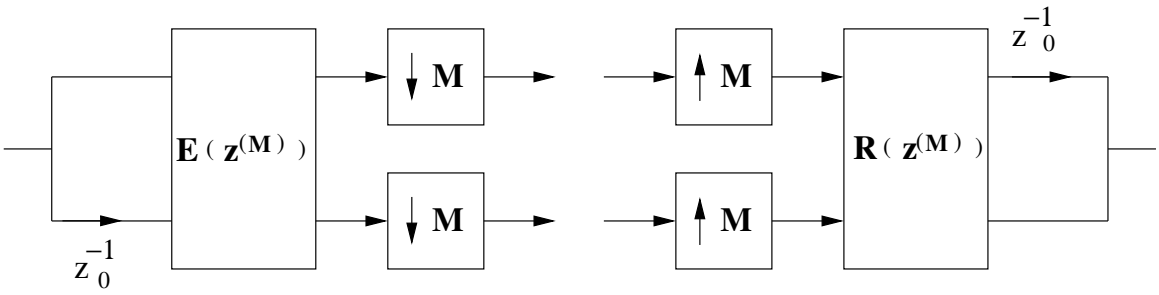


Figure 35: Non-separable structure for a maximally-decimated fan filter bank obtained by applying the multirate identities. An undecimated system is obtained by removing the sampling matrices **M**.

The development that follows relies on the multirate identity shown in Figure 34-(a). Here, an upsampler followed by the downsampler has no effect on the data, so $y[\mathbf{n}] = z[\mathbf{n}]$. However, for a quincunx resampling matrix $\mathbf{M}$, the intermediate upsampled signal $y[\mathbf{n}] = x[\mathbf{M}^{-1}\mathbf{n}]$ has gone through a 45 degree rotation.

Using this fact, we propose the following manipulation to the fan filter bank structure. First, consider the analysis stage. We have from identity and associativity that

$$\mathbf{M} = (\mathbf{M}^{-1}\mathbf{M})\mathbf{M} = \mathbf{M}^{-1}(\mathbf{M}\mathbf{M}) = \mathbf{M}^{-1}\mathbf{J},$$

where we define

$$\mathbf{J} = \mathbf{M}\mathbf{M} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}.$$

$\mathbf{J}$ is the well known rectangular downsampler where each dimension is downsampled separately by a factor of two.

In Figure 36 we present graphically the modifications to the fan filter bank structure. We assume the 2-D polyphase matrix $\mathbf{E}(\mathbf{z})$ represents any of the two approaches previously discussed in this work (i.e. the QMF or the ladder-based DFBs). So our development is general. Using the trick just described, we substitute the downsamplers by a chain of upsamplers followed by downsamplers as shown in Figure 36-(b). The downsamplers use the diagonal matrix $\mathbf{J}$. The next step consists of applying the identity of Figure 34-(e) to move the matrix $\mathbf{E}(\mathbf{z})$ to the left of the upsamplers. This is a relevant part of our derivation. Since $\mathbf{J}$ is a diagonal matrix with 2 as the diagonal entry, then the components of $\mathbf{E}(\mathbf{z})$ are upsampled by a factor of 2 in z_0 and z_1 independently. Since the entries of $\mathbf{E}(\mathbf{z})$, i.e., $\mathbf{E}_{i,j}(z_0, z_1)$ for $i, j = 0, 1$, are separable, then the elements of $\mathbf{E}(z_0^2, z_1^2)$ are also separable.

At this point we have decoupled the rotation and downsampling operations while the filtering operations remain separable without the need to downsample. The outputs of $\mathbf{E}(z_0^2, z_1^2)$ remain rotated and upsampled. As part of the features for the undecimated decomposition, we want the outputs to lie on the same sample lattice geometry as the input. At this point we exchange $\mathbf{J}$ by a couple of downsampling operators using the quincunx matrix $\mathbf{M}$, as presented in Figure 36-(d). The first downsampling operation returns the subband to the desired sampling lattice, while the second downsampler produces

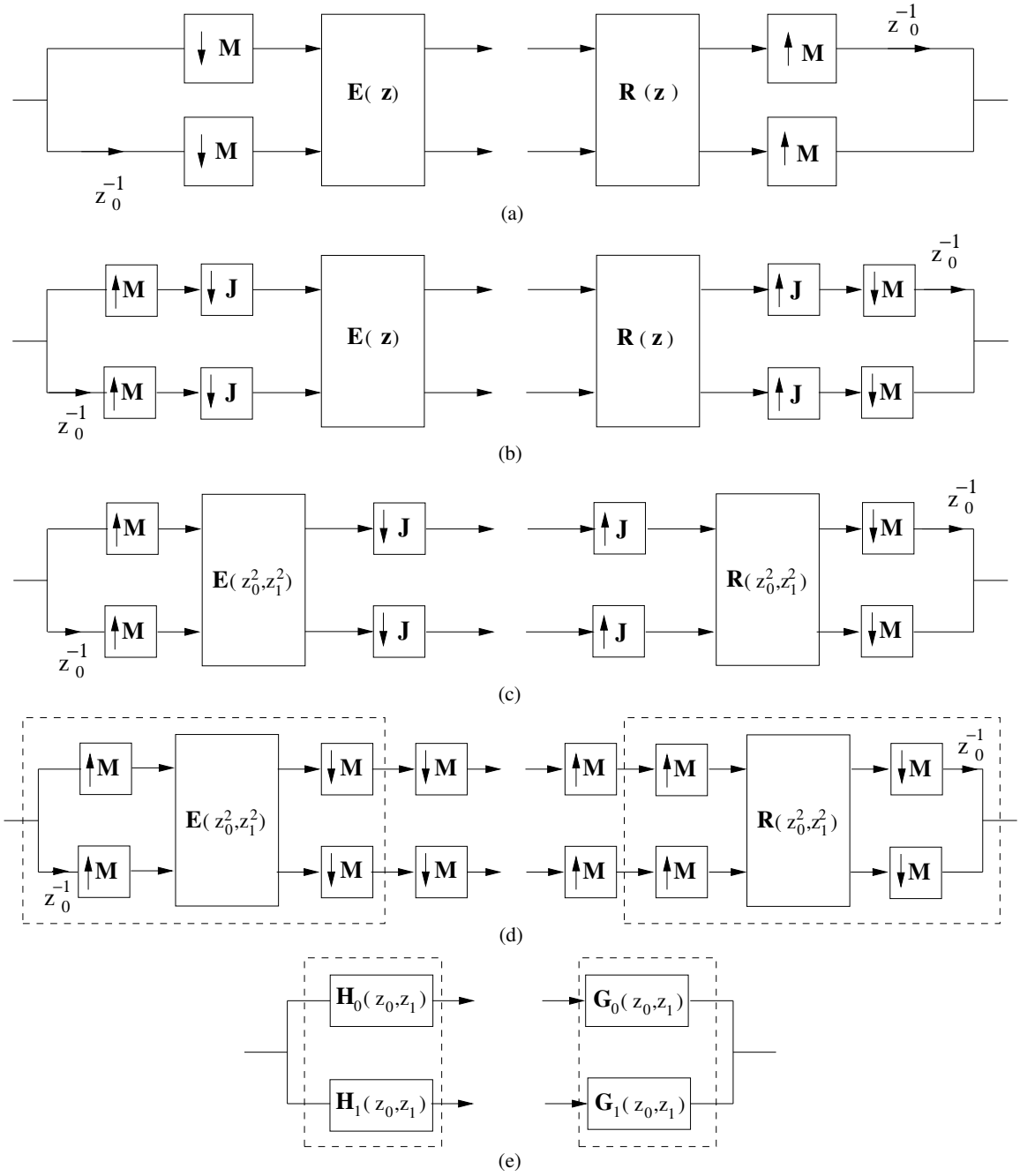


Figure 36: Graphical derivation of the low-complexity undecimated fan filter bank

the output for the maximally-decimated case. Obviously, for an undecimated structure we do not need the last downsampling operation. We have placed a dashed rectangle around the elements that confirm the undecimated fan filter bank (UFFB) structure. Effectively, the undecimated analysis stage implements the fan-shaped analysis filters $H_0(z_0, z_1)$ and $H_1(z_0, z_1)$ represented on the right side of Figure 36-(e).

Although our discussion has focused on the analysis stage, an analogous development can be made on the synthesis side as shown on the right side of Figure 36(a)-(d). In this case, the trick is reversed and upsampling by $\mathbf{M}$ is substituted by a chain of $\mathbf{J}$ upsampling followed by $\mathbf{M}$ downsampling. Then a similar reasoning follows by using the multirate property of Figure 36. A polyphase matrix $\mathbf{R}(z)_{up}$ with separable components is obtained.

Since for an undecimated decomposition the subbands have the same sample geometry and rate as the original data, there is a need for a rotation operation at the front end of the synthesis stage. This is easily achieved by changing the upsampler $\mathbf{J}$ with a chain of upsamplers using $\mathbf{M}$ as in Figure 36-(d). The first upsampler correspond to the maximally decimated structure and the second provides the required rotation for the undecimated fan filter bank. In Figure 36-(d) we enclose on a dashed rectangle the elements of the undecimated synthesis stage. The leftmost upsamplers are not needed in the undecimated case. Effectively, the undecimated synthesis stage implements the fan-shaped synthesis filters $G_0(z_0, z_1)$ and $G_1(z_0, z_1)$ as shown on the right side of Figure 36-(e).

For the case of ladder structures, the resulting UFFB analysis structure is shown in Figure 37. In this case we show the three-step ladder. The synthesis structure is straightforward to derive graphically.

2.4.3 The undecimated checkerboard filter bank

It is useful to look at the frequency domain to get some insight on the construction of the UFFB in Figure 36. We focus on the analysis stage. In Figure 38-(a) we show the unit frequency cell $[-\pi, \pi) \times [-\pi, \pi)$ emphasizing the desired fan-shaped partitions.

Upsampling by $\mathbf{M}$ generates a 45 degree rotation in the frequency domain and generates spectral images as seen in Figure 38-(b). This rearrangement of the frequency plane implies

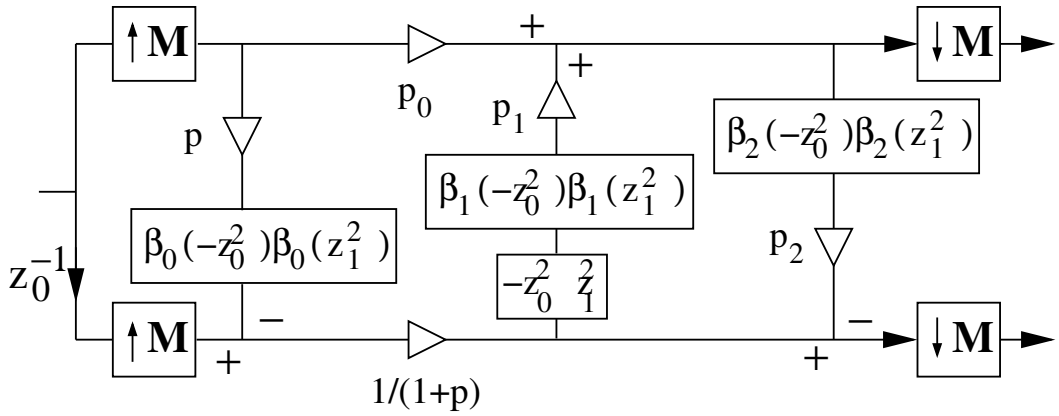


Figure 37: Ladder structure implementation of the UFFB

that the polyphase matrix $\mathbf{E}(z_0^2, z_1^2)$ must have the checkerboard magnitude responses shown in Figure 39-(a). The desired outputs are shown in Figure 38-(c). The final step is to downsample by **M** in order to undo the 45 degree rotation, and return the checkerboard regions into the desired fan-shaped passbands. An interesting point of our discussion above is the response of the matrix $\mathbf{E}(z_0^2, z_1^2)$ (Figure 39-(a)), which generates what we identify as two checkerboard responses. Hence we have the additional two-channel structure shown in Figure 39-(b). We identify the structure as the two-channel undecimated checkerboard filter bank (UCFB). The significance of this structure will be apparent in the next section describing the undecimated tree structured DFB.

2.4.4 Undecimated Tree-Structured DFB

In this subsection we show the construction of the undecimated tree-structured DFB. Similar to the maximally decimated case, it is possible to generate a 2^N-band DFB by cascading two-channel fan filter structures. However, the fact that no decimation takes place introduces some differences on the implementation.

Figure 40-(a) shows two frequency cells portraying the complementary support of the UFFB. Furthermore, these regions have been subdivided to represent the spectral partition for a four-band DFB. In order to generate these four regions, we need to apply the checkerboard filter bank to the frequency cells in Figure 40-(a).

Hence, the undecimated four-band DFB is implemented with an UFFB in cascade with

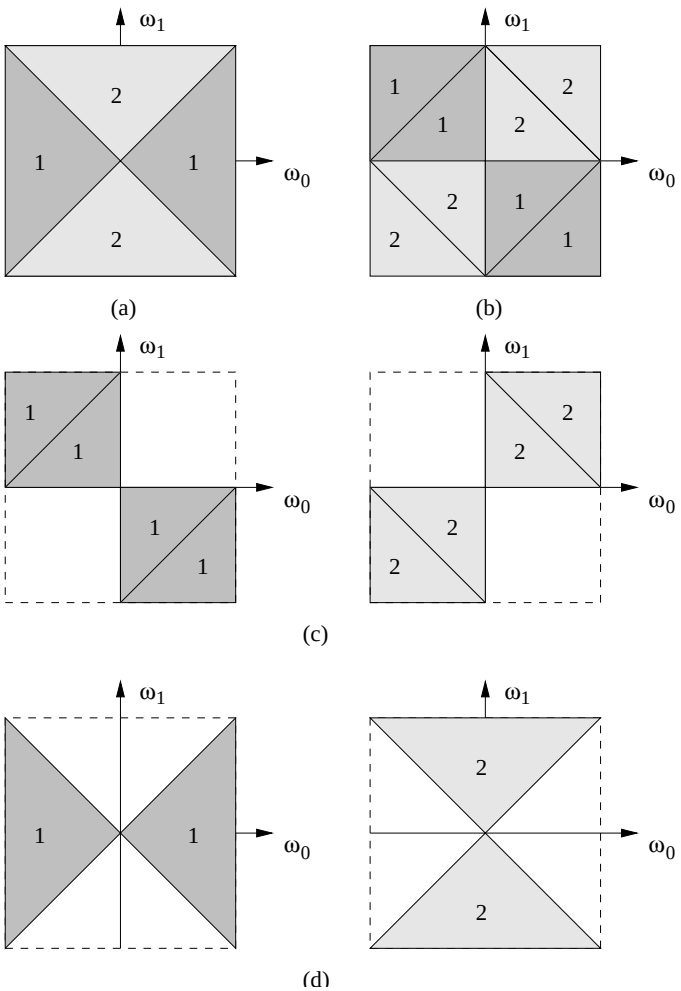


Figure 38: Graphical demonstration of the UDFB frequency domain processing. (a)Ideal fan-shaped partitioning of the frequency plane. (b)Modification of the fan-shaped partition after upsampling by **M**. Notice presence of images. (c)Outputs from $\mathbf{E}(z_0^2, z_1^2)$. (d)UDFB fan-shaped output after downsampling by **M**.

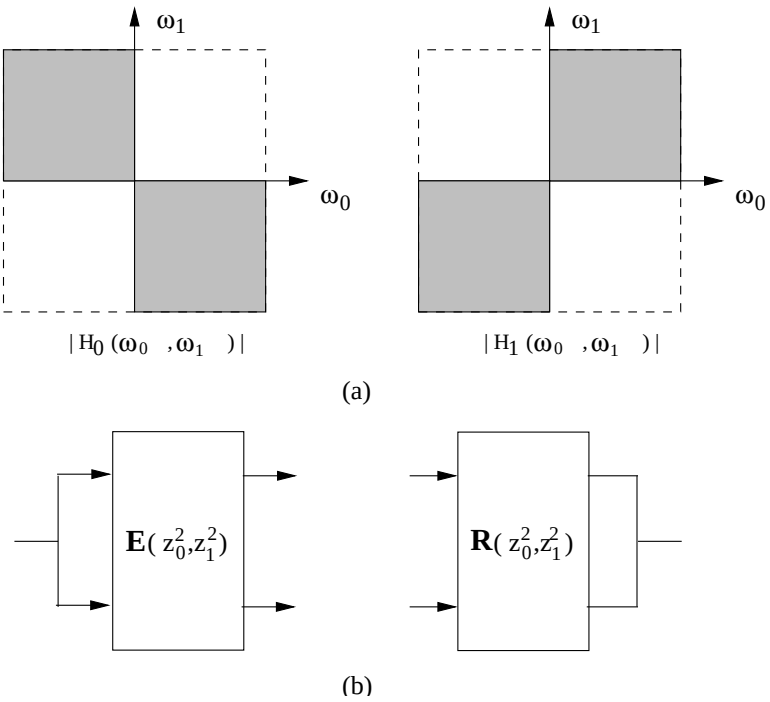


Figure 39: The undecimated checkerboard filter bank. (a) Inferred magnitude response for filters derived from $\mathbf{E}(z_0^2, z_1^2)$. (b) Two-channel undecimated filter bank structure.

two UCFBs shown in Figure 40-(b). As an example, we show the resulting four-band decomposition of our test image in Figure 41 using a three-step ladder structure with $\beta(z)$ designed with the parameters used for the 1-D responses on Figure 29.

Finally, we derive the structure for an eight-band DFB. As shown in Chapter 1, after the second stage we need to modify the sampling lattice using unimodular matrices $\mathbf{R}_i$. In Figure 40-(c) we depict one branch for the third stage. Resampling with matrices $\mathbf{R}_i$ modifies the 2-D frequency supports so that it is reshaped into a fan-shaped region without modifying the sampling rate. Then as with the second stage we need to apply the checkerboard filter bank to each output of the $\mathbf{R}_i$ resamplers. Finally, the corresponding backsampling matrix, $\mathbf{B}_i$, is used in order to reestablish the correct sampling geometry. In this case, the $\mathbf{B}_i$ matrix for the *ith* branch is $\mathbf{R}_i^{-1}$. For the eight-band DFB the four $\mathbf{R}_i$

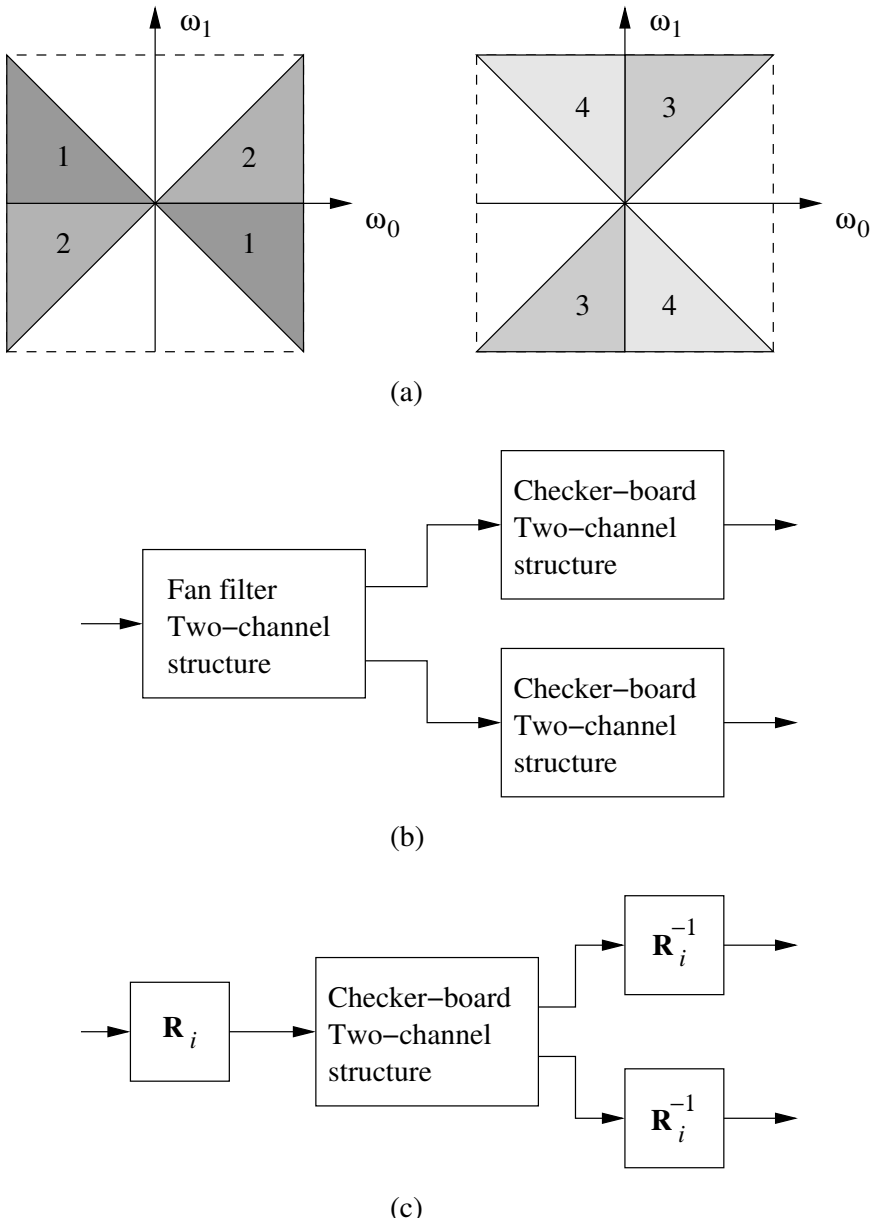


Figure 40: The undecimated DFB structure. (a) Four-band DFB frequency plane partition. (b) Tree-structure for a four-band DFB using fan filter bank and checkerboard filter bank.(c) Single tree branch for the third stage of an eight-band DFB.

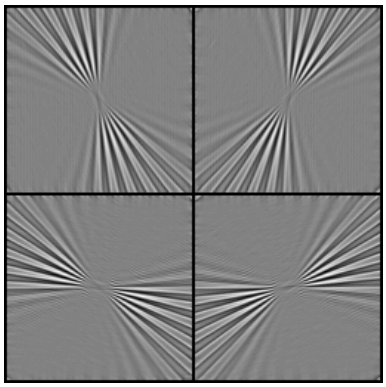


Figure 41: Example of a four-band undecimated directional decomposition. Each subband is the same size as the original image

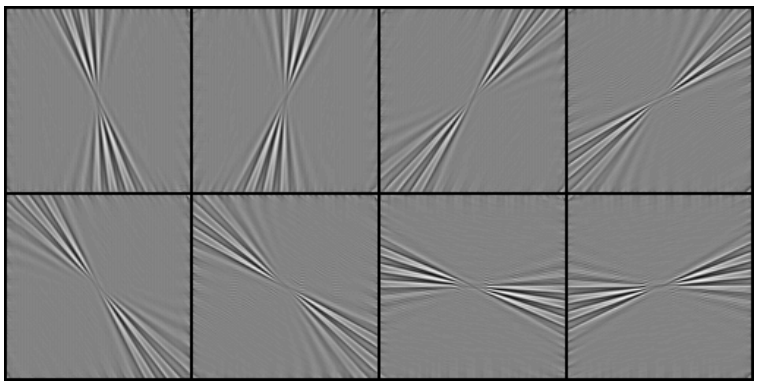


Figure 42: Example of an eight-band undecimated directional decomposition. The subbands are obtained by processing the four-band output on Figure 41

matrices are given by

$$\mathbf{R}_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad \mathbf{R}_2 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{R}_3 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \qquad \mathbf{R}_4 = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

We should note that other combinations of quincunx, $\mathbf{R}_i$, and $\mathbf{B}_i$ matrices are possible. As an example of an eight-band decomposition, we show the output of the system described above in Figure 42.

A different approach to get the third stage of the tree is to find resampling matrices

which rearrange the frequency spectrum so that they have checkerboard support. Then in this case, the UFFB structure has to be used after the second tree stage. However, the first approach is preferred since the number of required downsamplers and upsamplers is lower allowing for a simpler implementation.

2.5 Practical structures for the directional processing of images

In this thesis, processing of natural images will be a main area of interest. It is well known that natural images have their energy strongly concentrated at low frequencies. If we decomposed a natural image directly with the DFB or UDFB, the wedge-shaped passbands of the filters would introduce low frequency energy in the subbands. Since directional information is related to mid and high frequency information (like edges and textures), lowpass information can hinder our ability to capture and analyze these features. Moreover, the DFB is not capable of distinguishing between structures and detail at different scales. Hence, this limits the DFB ability to perform multiscale/multiresolution analysis.

To deal with these issues, in this section we present structures that combine the DFB and UDFB with image pyramids.

2.5.1 Simple Lowpass-Highpass decomposition

Even if we wanted lowpass energy on the directional subbands, the non-ideal nature of the filters would preclude us to do so. This non-ideal nature shows up most dramatically around the origin. As can be observed in Figure 32, some of the channels fall short of reaching the origin and the vertex is blunt. Other magnitude responses straddle the origin and the passbands are not sharp around the origin. Hence, DC information may be distributed unevenly across the bands. Since much of the directional information of interest resides on the mid to high frequencies, we frequently wish to have DC energy absent.

The simplest approach to deal with this issue is presented in Figure 43. We identify this structure as the "lowpass-highpass" decomposition. The image is first filtered with a lowpass filter $L_{\omega_c}(z_0, z_1)$ with cutoff frequency ω_c. To lower computational complexity, $L_{\omega_c}(z_0, z_1)$ can be computed with a row-column separable filter. The high frequency component is

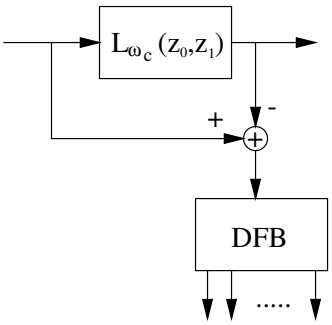


Figure 43: Lowpass-highpass analysis structure. The DFB or the UDFB can be used to do a directional decomposition.

obtained by a simple difference with the original image. Next, the directional analysis is performed only on the high frequency information. We note that in Figure 43, the residual is processed by the DFB, however we can also use the UDFB. It is easy to show that if no further processing is done on the subbands, the decomposition is invertible. In this structure, ω_c is an additional parameter we need to specify. The relevance of ω_c will be studied in the context of texture classification in Chapter 3.

2.5.2 Directional Pyramids

Other image decompositions like the 2-D DWT, the steerable pyramid [118], the complex-valued wavelet transform [69], and 2-D Gabor representations [11, 56] have an implicit multi-resolution (MR) or pyramidal component. This component could be an implicit part of the decomposition (e.g. the DWT), or could be implemented as a separate structure (e.g. the steerable pyramid [118]). In the latter case, we say that the decomposition is polar-separable, implying that a radial frequency decomposition (i.e. a pyramid) is performed independent of the angular decomposition. A comparison of the spectral partitioning between the DWT and a polar-separable structure is shown in Figure 47.

For image analysis and processing, scale information has been used in conjunction with directional information in applications where it might be necessary to distinguish objects or features of different sizes, for instance in image segmentation and texture synthesis. Moreover, these polar decompositions have been motivated by strong evidence that points (see [29] and references therein) to a similar kind of processing in the human visual system.

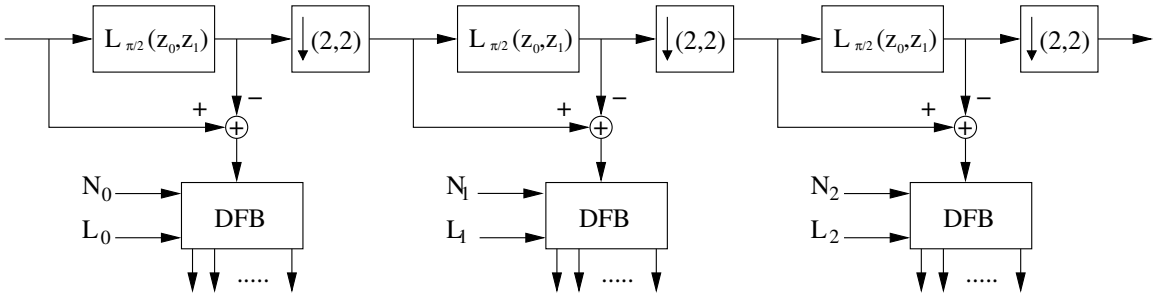


Figure 44: The Gaussian-pyramid structure combined with the DFB. Note, the DFB can be substituted by the UDFB.

It will be highly desirable to have such type of representation in our case.

The Bamberger DFB is purely directional. Thus, we have the advantage to select the radial decomposition of our choice. Next, we introduce DFB- and UDFB-based pyramidal structures. We will make extensive use of them in subsequent chapters. We will refer to these structures as Directional Pyramids (DPs). Without loss of generality, we present the DPs using the maximally-decimated DFB, however it is easy to see that the DFB can be easily replaced by the UDFB.

Depending on whether an image needs to be reconstructed or not we have a couple of choices for the radial decomposition. If reversibility is not an issue, we can use a modified Gaussian pyramid, as shown in Figure 44. In this case, the detail image is obtained as the difference of the lowpass filtered image with the input image for a corresponding resolution level. At each pyramid level, the number of directional bands N and the length L of the $\beta(z)$ functions can be selected independently of the others.

Another straight forward approach is to take the output of a Laplacian or Laplacian-like pyramid [14] structure and further decompose the detail subbands with the DFB. The analysis structure is presented in figure 45. It is easy to show the reversibility of this representation.

These two structures are overcomplete with an increase in data by approximately a factor of $\frac{4}{3}$. We refer to them as the Gaussian-DFB pyramid and the Laplacian-DFB pyramid. For the cases where the UDFB is used, the increase in data is significantly larger. For the

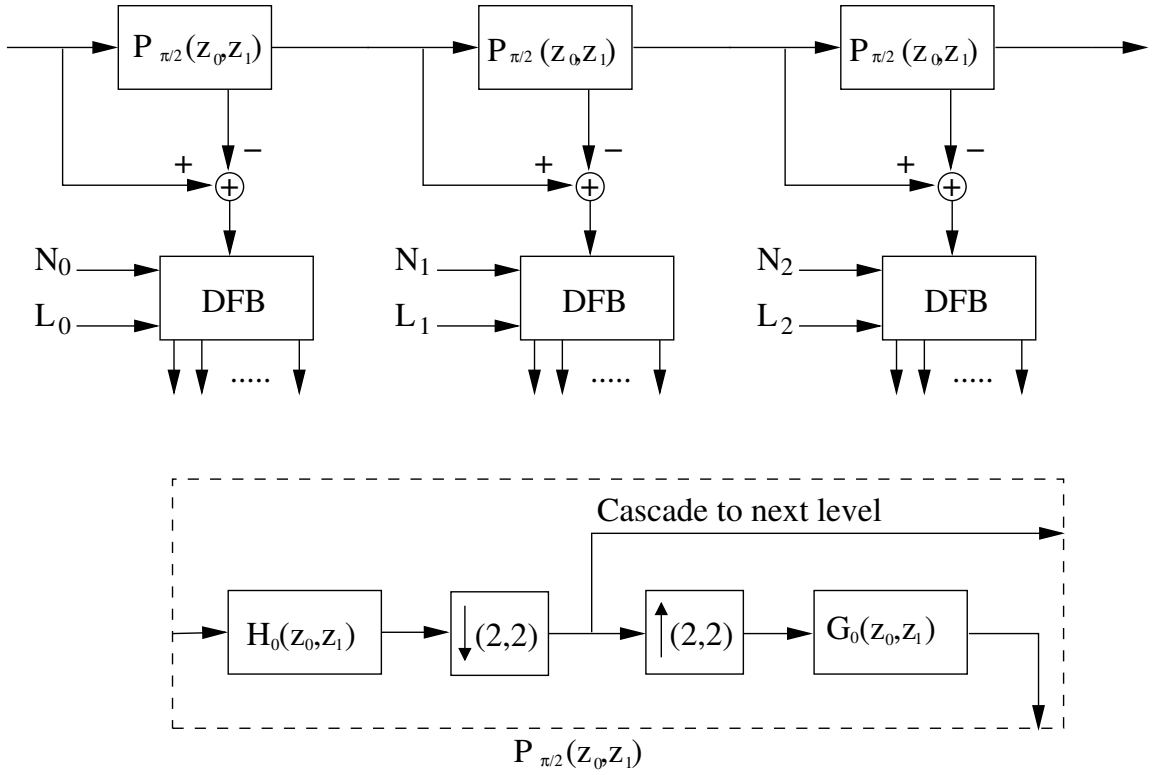


Figure 45: The Laplacian pyramid structure combined with the DFB.

case where all radial bands are decomposed into an N-band UDFB, the data increase is given by a factor of $\frac{4N}{3}$. We refer to these structures as the Gaussian-UDFB pyramid and the Laplacian-UDFB pyramid.

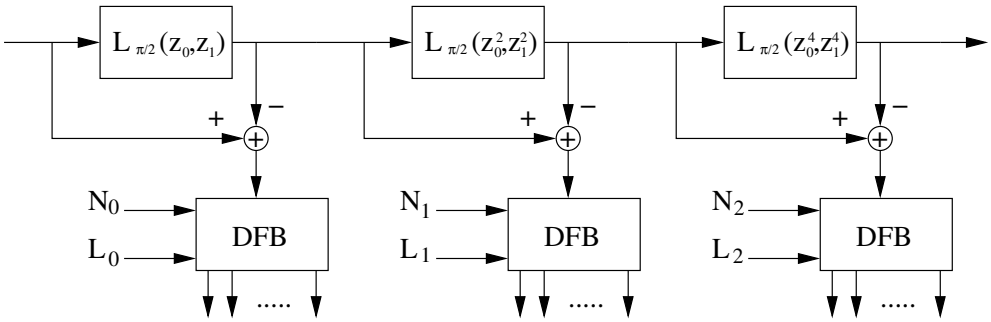
There is a possible drawback on using these structures. Each detail level D_j of the pyramid is one quarter the size of the previous level D_{j-1}. In the case of the N-band DFB, the directional subbands are $\frac{1}{N}$ the size of D_j. The tradeoff between spatial and angular resolution might render the directional subbands at the coarser resolutions useless for further processing and/or for accurate calculation of statistics.

To alleviate this drawback, we propose structures in which the radial components are not downsampled. We show the modified systems in Figure 46. In order to eliminate the downsamplers, we apply multirate identities so that they can be phased-out to the left of the structures in Figure 44 and 45. Similar to the undecimated DWT, we use one single lowpass prototype $L_{\frac{\pi}{2}}(z_0, z_1)$ which is modified for each resolution level as $L_{\frac{\pi}{2}}(z_0^{2^j}, z_1^{2^j})$ where $j = 0, 1, ...J-1$. In the spatial domain this modification corresponds to inserting zeros in the impulse response of $L_{\frac{\pi}{2}}(z_0, z_1)$. Hence, the radial decompositions are identified as the Undecimated Gaussian (UGaussian) pyramid, and the Undecimated Laplacian (ULaplacian) pyramid.

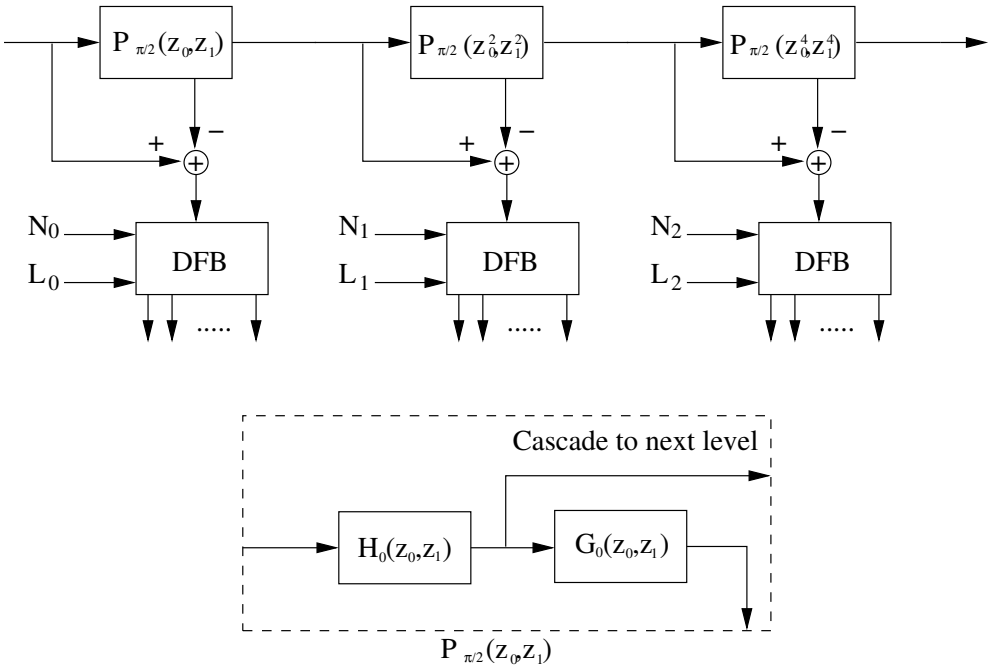
When combined with the DFB, the resulting DPs are invertible and are overcomplete by a factor of J. If the UDFB is used, the data increases by a factor of $N(J-1)+1$. We identify four more DPs for this configuration: the UGauss-DFB pyramid, the ULap-DFB pyramid, the UGauss-UDFB pyramid, and the ULap-UDFB.

If we use the same number of directional bands for all J pyramid levels, all the subbands corresponding to the same orientation will have the same size. This is a nice feature which allows us to establish a one-to-one correspondence between subband coefficients at different resolutions, eliminating parent-child ambiguities for inter-subband processing.

A comparison of the frequency plane partitioning obtained with the DP structures and the traditional separable 2-D DWT is shown in Figure 47. We see that the 2-D DWT has limited angular sensitivity (diagonal, horizontal and vertical directions), while the DP structures can be adapted to have 2^n directional subbands. However, it is important to note

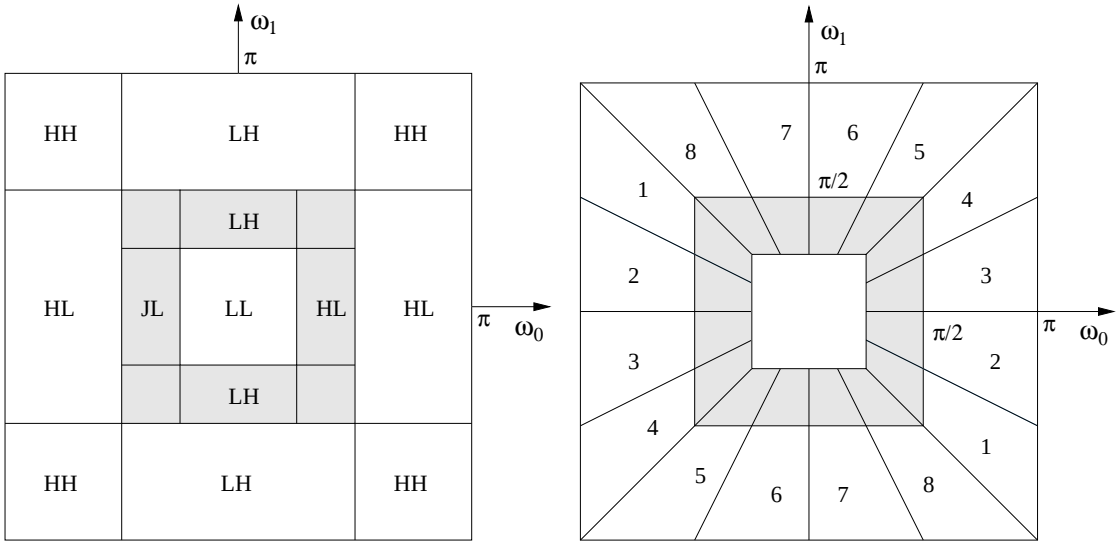


(a) Undecimated Gaussian-DFB pyramid



(b) Undecimated Laplacian-DFB pyramid

Figure 46: Invertible DPs structures with no decimation of the resolution components.



(a) 2-D separable DWT (b) Directional Pyramid

Figure 47: Comparison of the frequency plane partitioning of the 2-D separable DWT and the proposed directional pyramids.

that the DWT is critically sampled, while DPs be moderately to substantially overcomplete, depending on which DP structure we select.

As an example, we show the Laplacian-UDFB pyramid for the test image from Figure 11. We let $J = 3$ and $N = 4$ (note that the coarsest level consists only of a lowpass subband).

An additional directional pyramid based on the radial component of the steerable pyramid [118] will be introduced in Chapter 5 for the study of image denoising.

2.5.3 Further image representations using directional decomposition structures

Our ability to control the radial bandwidth and orientation sensitivity of the decomposition separately allows us to generate a large variety of more "exotic" radial-directional filter bank structures. We cite two possible structures which are potentially useful and should be object of future study:

1. Uniform-radial, uniform-angular decomposition. The frequency plane partitioning is presented in Figure 49 and can be thought of as analogous to a 2-D separable uniform

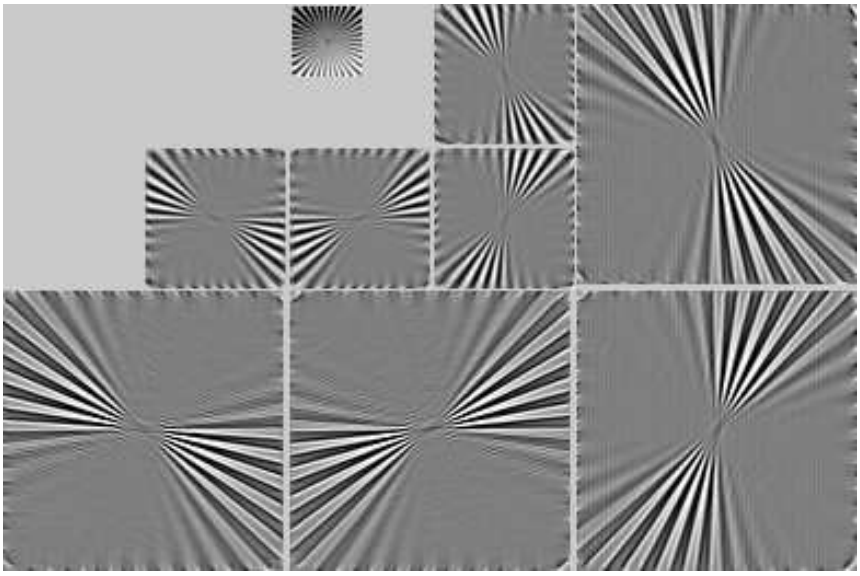


Figure 48: Laplacian-UDFB pyramid example with $J = 3$ and $N = 4$.

filter bank. In this case, the passband sizes in the radial direction are the same as opposed to dyadic. Any of the pyramid structures presented above (decimated and undecimated) might be modified to this decomposition.

2. Directional packets. Similar to the idea of wavelet packets (i.e. adaptive non-uniform filter banks [19]), there can be 2-D signals whose energy is concentrated along specific orientations. Hence, a scheme that modifies the directional tree-structure based on an energy criterion can be adopted. Obviously, this structure could be combined with a pyramidal component.

2.6 Chapter Summary

In this chapter we have proposed three contributions to the theory of the Bamberger DFB. First, we have shown that ladder structures can be used as the filtering polyphase structure. This structure leads to PR-FIR fan and diamond filters, which had not been used before in the Bamberger DFB context. The ladder structure is still separable in the polyphase domain, and allows for different FIR filter design methods and nonlinear operations to be used easily.

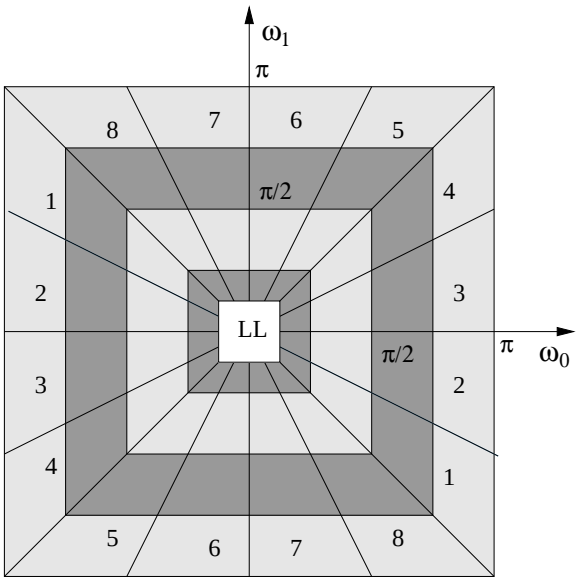


Figure 49: Uniform partitioning of frequency plane with a combination of concentric frame-shaped filters and the DFB.

Second, we have presented an undecimated DFB that is obtained by simple manipulations of the resampling matrices. This undecimated decomposition is useful for applications where shift-invariance is desired.

Finally, we have adapted pyramid components to the DFB and UDFB. We have presented different decomposition structures that fit different application requirements. We have dubbed such representations as directional pyramids. Their applicability will become apparent in subsequent chapters.

A lot of attention thus far has been given to the DFB structure and its properties. However, there has been a limited amount of work on applications of this decomposition to image processing and image analysis problems. Moreover, no comparisons with other image representations have been reported. In the subsequent chapters we do an extensive study on texture classification, texture segmentation, texture synthesis, and image denoising. This study reports a detail performance comparison with other image decompositions.

CHAPTER III

MULTICHANNEL TEXTURE ANALYSIS AND CLASSIFICATION WITH BIORTHOGONAL DIRECTIONAL FILTER BANKS

3.1 The concept of texture

Early research on the understanding of the mechanisms that take place in our visual system was presented by Julez [63, 64]. At the time, Julez conjectured that textures could be recognized and differentiated solely through their first and second order statistics. Although this conjecture was later proven wrong by Gagalowicz and Ma [41], it helped spark the large effort on texture research as a branch of image analysis and computer vision.

It is well recognized that texture is an important categorization in terms of recognition and interpretation of visual information. However, to date there is no uniformly accepted definition of texture. In many cases this definition depends on the particular application. But generally speaking, we might say texture is related to the consistency of the surface of materials that form an object or scene. Hence, the tactile perception of coarseness or smoothness of a surface can be translated visually as a property of graininess or patterning. The tonal variations can go from random to highly structured.

Properties of texture have been studied in the context of computer vision. While a comprehensive and accepted method of texture description is not yet available, it is insightful to examine some of the descriptions that have been proposed previously.

Neighborhood properties of a pixel can provide a useful characterization of some textures [126]. Depending on the relationships among such neighborhoods, textures may be classified as *stochastic* or *structural* [47]. Stochastic textures are those whose elements do not seem to follow a specific placement order; the appearance of the texture derives from the random

combination of these elements. In contrast, structural textures manifest periodic and quasi-periodic properties. Texture properties are typically global in nature [56].

Many textures can be considered to have both structural and stochastic components. In [121], six features common to all textures were proposed: coarseness, contrast, directionality, line-likeness, regularity, and roughness. A more compact set of features were considered in [38, 76], where directionality, periodicity, and randomness are considered as the most significant features. An objective measure of perceptual texture features is desirable in order to use texture for the analysis and understanding of images. In order to arrive at useful quantitative perceptual measures described above, it is necessary to consider the application or problem to solve. We can identify three general problems that have been extensively studied:

- **Texture Classification:** Given a set of N different texture classes, the problem is to extract a set of features useful to assign a new texture sample to one of these classes.
- **Texture Segmentation:** Here, we use the textural features of an image to segment the image into regions with homogeneous texture. Again it is necessary to extract features capable of assigning each pixel to a texture class. Some related problems are the accurate estimation of the boundaries among the textured areas, and the estimation of the number of textures present in the image.
- **Texture Synthesis:** Here the objective is to obtain a model of the texture, such that the parameters of the model can be used to generate a new sample which is visually indistinguishable from the original.

Next, we give a brief review of the previous work reported in the literature to tackle these problems.

3.2 Review of previous work on texture

A good review of early work on texture research is summarized in the 1979 paper by Halarick [47]. Depending on the characteristics of the texture, statistical and structural approaches were described.

The early statistical methods extracted textural information through a variety of mathematical tools, like the autocorrelation function, optical processing, digital transforms (Fourier, Hadamard, Slant), gradient information (i.e. edgeness), morphology, co-occurrence matrices, grayscale spatial dependence models, and autoregressive models. Many of these approaches were ad-hoc, and emerged later under the more formal frameworks.

The structural approaches were mostly based on the extraction of texture primitives and a set of placement rules [78, 47, 23, 84, 83, 123]. The primitives are distinctive elements of a texture which follow a certain order defined by the placement rules. Texture where placement rules are well defined are identified as *strong textures*, while the more random looking textures are labelled as *weak textures*. Weak textures are easier to model with statistical approaches.

In the seventies and the eighties, stochastic texture research was mostly based on spatial interaction models such as autoregressive (AR) models [66, 67, 81, 37] and Markov models [10, 30, 48, 25, 53]. The coefficients of the model are then used as features for analysis or synthesis. A lot of the work on these model was directed at the estimation of the model parameters and the order of the model. More recent work in this area by Comer and Delp [22] has combined multiresolution analysis with AR models. However, these models are only good at capturing short range relationships and have failed to give good models of structural textures.

More recent models based on Gibbs distributions and Markov chains were explored. In particular, in [142] the idea of Maximum Entropy was used to derive an elaborate stochastic model of texture for statistical and highly structured textures. Finally, Francos, et al., [38] introduced the 2-D Wold decomposition to model both structural and stochastic textures by separating periodic, directional, and stochastic features.

3.2.1 The Multichannel Paradigm for Texture Analysis

A different class of tools is based on multichannel analysis. The method extracts information by passing the texture sample through a bank of filters. The output of each filter (i.e. a channel) provides information about different features of the texture.

Perhaps the first manifestation of this paradigm was presented by Laws [72]. He used a set of small 2-D separable kernels to extract information about the edgeness, spotness, and flatness properties of the texture. Next, energy of the channels was estimated and grouped to form feature vectors. The insight for using energy as a feature comes from the view that similar textures should exhibit similar spectral characteristics which can be captured by the energy of each channel. The combination of these energies gives a signature for the texture. Laws used 25 masks (i.e., filters) in his experiments, and achieved good results when compared to other methods. The significance of Laws work is that it showed a very simple scheme that often outperformed more complex approaches like the AR and Markov models.

Unser [126] showed that *local spatial transforms* could fit Laws' scheme and improved the classification performance. He experimented with the KLT, DCT, DST, and Hadamard transforms. As features he used the histogram and energies of the transform coefficients. Local transforms can be viewed as a multichannel scheme by considering each transform coefficient as a separate channel. In general, it has been seen that multichannel methods outperform second order statistics methods and spatial interaction models while being less computationally expensive [19, 46].

There is a strong psychovisual justification for the multichannel paradigm. The human visual system (HVS) behaves like a set of orientation sensitive bandpass filters. This conclusion was determined experimentally by registering the reaction of test subjects to sinusoidal gratings. De Valois, et. al, [131] measured the *receptive fields* (i.e. frequency responses) of the macaque's visual cortex when they were subjected to sinusoidal grating stimuli. They found that cells respond to a narrow frequency range and orientation. Experimental reports suggest that we have sensitivity to about 30 orientations.

Multichannel processing is a "divide and conquer" approach, where we are able to use simpler analysis techniques on each channel, and extract independent features that can then be used for analysis.

The receptive fields measured in [28] resemble 2-D Gabor functions (i.e., modulated Gaussians) and have been used to develop different image decomposition techniques. Bovik,

et al. [11] used complex 2-D Gabor functions oriented at different angles and different scales for texture analysis and segmentation. Jain and Farrokhnia [56] developed a texture segmentation algorithm using real even-symmetric Gabor functions. Haley and Manjunath [46] have proposed a polar complex Gabor representation that has been used for rotation invariant texture classification.

A major issue with Gabor filters is how to discretize the continuous functions, so that aliasing is minimized when working in the discrete domain. Usually systems of this type have high sampling rates, are computationally intensive, and have large storage requirements.

On a parallel track, the concept of filter banks and wavelet transforms were under study in the signal processing and mathematics communities. In particular, the discrete wavelet transform (DWT) has been extensively studied.

The generalization of filter banks and the DWT to 2-D is done by forming 2-D separable basis. Most of the research on the 2-D DWT has focused in image coding; however, the connection to the texture problem and the multichannel paradigm was also established and investigated in [127, 35, 102].

In contrast to Gabor representations, subbands (i.e. channels) are maximally decimated which implies shift invariance is lost. This could be problematic for pattern recognition problems. Finally, the 2-D DWT has limited angular selectivity along horizontal, vertical and diagonal orientations.

To overcome some of the short comings of the 2-D DWT, other related 2-D filter bank representations have been reported. Unser dealt with the issue of shift invariance in [128] by using wavelet frames (also known as the undecimated DWT). From the implementation point of view, it consists of an dyadic filter bank without downsamplers, hence all the subbands are the same size as the input image. The memory and computational requirements increase by a factor of $3L + 1$, where L is the number of decomposition levels. Unser reports good results for the case of texture classification and segmentation. An interesting conclusion reached in his work is that the regularity (i.e. smoothness) of the wavelet is not a relevant feature in the classification case.

The 2-D DWT has a fixed frequency domain partitioning which does not correspond

to the energy distribution of most textures. An adaptive decomposition was proposed by Chang and Kuo in [19] for texture classification. They proposed a Tree Structured Wavelet Transforms (TSWT), which is also known as Wavelet Packets. The idea is to form a non-uniform filter bank decomposition by adaptively decomposing the image according to some energy or information theory criterion (e.g., entropy). The decomposition remains maximally decimated, however the extra step of computing the subband energies adds computational complexity to the procedure. Chang and Kuo achieve excellent classification results using the TSWT. In most cases the TSWT outperforms other methods reported in the literature.

Both the wavelet frames and TSWT representations used the fixed separable decomposition. However, by using the separable decomposition we are forced to give away the possibility of having a finer degree of orientation selectivity in our decomposition. This is of particular interest for textures since they embody features that are oriented at arbitrary angles. Motivated by the importance of directional information, a few orientation selective non-separable pyramids have been proposed in the past, for instance, schemes based in Gabor decompositions.

Vautrot, et al. [132], used the 2-D continuous wavelet transform for texture classification and segmentation. In particular they studied the generation of rotationally invariant feature sets. As with Gabor functions, the discretized functions are non-separable and require significant computation.

The steerable pyramid from Simoncelli, et al. [118] has the ability to decompose a signal into multiresolution directional subbands. The decomposition has been used to successfully build stochastic models for texture synthesis [103]. This representation is overcomplete by a factor of $4k/3$, where k is the number of orientations.

Other recent ideas on generating 2-D oriented representations have been the brushlets proposed by Coifman and Meyer [86], and the so called curvelets reported by Candes and Donoho [15]; however, no work on texture research has been reported for these tools.

Finally Kingsbury [69] has proposed the Dual Tree Complex Wavelet Transform (DT-CWT). This filter bank consists of 2 dyadic-tree filter banks, one serving as a real channel

and the other as an imaginary channel. When the trees are combined, they provide an analytic representation for a signal. Hence, for the 2-D case, it is possible to separate information along six orientations ($\pm 15^\circ$, $\pm 45^\circ$, $\pm 75^\circ$). This representation is near shift invariant and has low computational complexity (i.e., the filtering operations remain separable), but redundancy by a factor of four is introduced. The 2-D DT-CWT has been used with good results for texture classification [51] and texture segmentation [109].

As is evident from this overview, the directional filter bank (DFB) developed in the previous chapter is one more option to the multichannel paradigm. In this and the next chapters we explore solutions to texture analysis and synthesis problems using the DFB. As we go into the chapter, we characterize the DFB with respect to computational complexity, memory requirements, shift variance, directional selectivity, passband quality, and other aspects unique to the DFB.

3.3 General Scheme for Multichannel Texture Classification

For the case of texture classification, we can identify three stages for implementing a system:

- Image Transformation: Generates a set of N channels (i.e., subbands) which capture different characteristics of the texture.
- Feature Extraction: An energy measure is calculated for each channel. The energies are grouped into an N-dimensional feature vector.
- Classifier: Each feature vector is assigned to one of K classes using a classification rule.

We have covered the multichannel transformations on the previous section. Next we discuss the selection of the features and the classifier, and determine the feature set and classifier to be used in conjunction with the DFB.

3.3.1 Feature Selection and Extraction

Textures have energy concentrated at specific spectral regions depending on their particular visual characteristics. Hence, measurement of the channel energies has been used as very simple characterization of the texture.

Unser [126], presents the design of an optimal texture classification transform that maximizes the difference in energies of the channels. As would be expected, the optimal transform is signal dependant (the spatial covariance matrices are needed), and it is related to the well known Karhunen-Loeve transform (KLT). In practice, optimal transforms are expensive to calculate and could change for different samples of the same texture. Similar work for the case of two textures has been carried by Randen and Husøy [107]. Their objective was to design an optimized filter bank for texture segmentation. Hence, for practical implementations the energy from suboptimal multichannel representations discussed are used as features.

From the statistical point of view, each channel is a realization of a zero-mean random field. If the channel pdfs have normal distributions, a sufficient characterization of the texture could be obtained in terms of the channel variances given by

$$\sigma_i^2 = \mathrm{Var}\{y_i\} = E\{(y_i - E\{y_i\})^2\}, \tag{41}$$

where y_i represents the transform ith channel, $E\{\cdot\}$ is the expectation operator, and in our case $E\{y_i\} = 0$. Looking at the variance as a measure proportional to signal energy is reminiscent of the ℓ_2 norm given by

$$\|x\|_2 = (\sum_k |x_k|^2)^{\frac{1}{2}}. \tag{42}$$

This is particularly true if we use the maximum likelihood (ML) unbiased estimate of channel variance

$$\sigma_i^2 = \frac{1}{N_1 N_2 - 1} \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} y_i(n_1, n_2). \tag{43}$$

If in addition, the channels are decorrelated (i.e. the transform is orthogonal), then the energy separation is maximized.

In general, the orthogonality and Gaussian assumptions are not satisfied. Nonetheless, channel variance has been used as a measure of energy. The good to excellent performance of these systems validates the intuition of using energy based features. An alternative to modelling the pdfs as Gaussian would be to use higher order moments (i.e., skewness, kurtosis, etc.). Experimental evidence presented in [126] suggest that they could improve classification in some cases, but generally the improvements are marginal.

If the texture sample contains inhomogeneities caused by the acquisition process (i.e., a camera, scanner, or any imaging device), or because the texture itself presents anomalies (like shadows, fabric blemishes, etc.), then the estimate in Equation (43) could be biased by the presence of outliers causing deterioration of classifier performance.

Hence a variance estimator that can handle outliers distortion is desirable. A robust estimate for σ is given by

$$\hat{\sigma}_{robust} = \frac{1}{0.6745}\text{MED}(|y_i|), \tag{44}$$

where the MED operator calculates the median value of the coefficient magnitudes of subband y_i, and the pdf of y_i is assumed to be Gaussian. This estimate has been popular in wavelet denoising applications as discussed in Chapter 5.

Another measure of energy is the ℓ_1 norm. For the case of discrete signals, the ℓ_1 norm defined by

$$\|x\|_1 = \sum_k |x_k|. \tag{45}$$

Chang and Kuo [19], have evaluated most of the well known multichannel systems using the ℓ_1 norm as a feature. They reported little difference in performance between the ℓ_1 and ℓ_2 norms, while big computational savings are provide by the former. They used the ℓ_1 norm estimate given by

$$e_1 = \frac{1}{N_1 N_2}\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}|y_i(n_1, n_2)|, \tag{46}$$

where y_i is a ith channel of size $N_1 \times N_2$.

For subsequent use we identify the variance estimates as energies by defining

$$e_2 = \hat{\sigma}^2_{unbiased}, \tag{47}$$

and

$$e_3 = \hat{\sigma}^2_{robust}. \tag{48}$$

In this thesis, we choose e_1, e_2, and e_3 as features to characterize a texture. For each energy measure e_i, we form feature vectors by grouping the energy estimate of each channel on an N-dimensional feature vector

$$\mathbf{e}_i = [\ e_{i,1} \quad e_{i,2} \quad \ldots \quad e_{i,N}\ ]^T, \tag{49}$$

where $i = 1, 2, 3$. The ordering of the features $e_{i,j}$ within a vector corresponds to the subband numbering used in Figure 2 for either $N = 4$ or $N = 8$. These orderings are arbitrary. In subsequent sections, we experimentally assess the discriminative power of the proposed features for texture classification.

3.3.2 Selection of a Classification Scheme

Finally, we need to select a scheme to assign each feature vector to a texture class. A classification rule or procedure needs to be selected. Different types of classifiers have been studied in the literature of texture classification. These include clustering schemes, minimum distance classifiers, neural networks, and fuzzy methods among others. Also, classifiers can be parametric or non-parametric.

In our case, we will use a parametric scheme based on the Bayes distance. The Bayes distance classifier has been a popular choice in texture research [126, 128, 46]. In [19] this distance outperform other distances (e.g. Euclidean, Mahanalobis, etc.) for a set of representative multichannel transformations. Hence, we also will be able to do an "apples-to-apples"comparison within the DFB-based feature set and the other systems.

The Bayes distance between texture class k and a feature vector $\mathbf{v}$ is defined as

$$d_k(\mathbf{v}) = (\mathbf{v} - \mathbf{m}_k)^T \mathbf{C}_k^{-1} (\mathbf{v} - \mathbf{m}_k) + \log(\det(\mathbf{C}_k)). \tag{50}$$

The distance requires the estimation of the mean feature vector $\mathbf{m}_k$ and covariance matrix $\mathbf{C}_k$ for each texture class. Hence, it is assumed that the number of classes, K, is fixed, and that there is a set of training patterns to estimate the parameters.

For the kth class, parameter estimation is carried out using the maximum likelihood (ML) estimators on the set of N-dimensional training vectors $\{\mathbf{v}_{k,j}\}_{j=1}^{J}$, namely,

$$\hat{\mathbf{m}}_k = \frac{1}{J} \sum_{j=1}^{J} \mathbf{v}_{k,j} \tag{51}$$

$$\hat{\mathbf{C}}_{k,p,q} = \frac{1}{J} \sum_{j=1}^{J} (\mathbf{v}_{k,j}[p] - \mathbf{m}_k[p])(\mathbf{v}_{k,j}[q] - \mathbf{m}_k[q]), \tag{52}$$

where $\hat{\mathbf{C}}_{k,p,q}$ corresponds to the $(p,q)th$ entry on the covariance matrix. We have used the notation $\mathbf{v}_j[p]$ to address the pth feature from the jth training vector, and $\mathbf{m}_k[p]$ to address

the *pth* element of the mean feature vector.

Once $\mathbf{C}_k$ and $\mathbf{m}_k$ are estimated for each class, classification can be performed on test vectors by assigning test vector $\mathbf{v}$ to class k according to the following rule

$$k^* = \arg\min_k \{d_k(\mathbf{v})\} \tag{53}$$

for $k = 1, 2, \ldots, K$. In particular, if the conditional pdfs of the feature vectors is multivariate i.i.d. Gaussian for each class, then the Bayes distance of Equation (50) is equivalent to the minimum error Bayes classifier.

3.4 A Novel Multichannel Texture Classification Algorithm using the Biorthogonal DFB

The developments and results of this and the following sections present original research in the topic of texture classification using the biorthogonal directional filter bank.

We use the ladder-based DFB described in the Chapter 2 to generate the multichannel representation. We use a tree-structured decomposition with the two-channel ladder structure as in Figure 31. To avoid lowpass leakage, we use the low-high structure of Figure 43. The selection of this configuration is arbitrary, but the experimental results show this is a good choice. Also, the two-step ladder structure is used, as there was no performance advantage in choosing the thee-step ladder structure.

In summary, the texture classifier first separates a highpass component from each (training and testing) sample. Then the highpass band is analyzed with an N-band biorthogonal DFB. The energy of each subband is estimated and grouped in an feature vector $\mathbf{v}$. The training patterns are used to estimate $\mathbf{m}_k$ and Ck. The test vectors are assigned to the texture class with the smallest Bayes distance [Equations (50) and (53)].

There are a number of defining parameters to consider. Namely, we have

- The cutoff frequency ω_c of $L_{\omega_c}(\omega_0, \omega_1)$ in Figure 43.
- The length L of the ladder stage filters $\beta(z)$.
- The number of directional bands N.

Table 1: Textures used in experiments from [19]

Texture Label	Texture Description	Texture Label	Texture Description
D3	Reptile skin	D57	Handmade paper
D4	Pressed cork	D65	Handwoven oriental rattan
D6	Woven aluminium wire	D68	Wood grain
D9	Grass Lawn	D74	Coffee beans
D11	Homespun woolen cloth	D77	Cotton canvas
D16	Herringbone	D78	Oriental straw cloth 3
D19	Woolen cloth with soft tufts	D79	Oriental grass fiber cloth
D21	French canvas	D82	Oriental straw cloth 4
D24	Pressed calf leather	D83	Woven matting
D29	Beach sand	D84	Raffia looped to a high pile
D34	Netting	D92	Pigskin
D36	Lizard skin	D95	Brick wall
D52	Oriental straw cloth 1	D102	Cane
D53	Oriental straw cloth 2	D103	Loose burlap
D55	Straw matting	D105	Cheesecloth

- The design method of $\beta(z)$. In particular, we will compare maximally-flat designs vs. Parks-McClellan filters.

In the next sections, we do an experimental evaluation of the proposed classification scheme. In particular, we explore the impact of sample size on system performance. Smaller texture samples introduce more statistical variation in the features, making parameter estimates less reliable.

3.4.1 Evaluation of the classifier

The choices we have made so far are closely related to the work of Chang and Kuo [19]. Our purpose is to compare the performance of the DFB with other multichannel transforms. In this subsection we evaluate the classifier using the same set of 30 textures from the Brodatz album [12]. This set provides a good representation of different types of textures. A description of the textures is listed in Table 1 and sample patches are shown in Figure 50.

Our first set of experiments consists of extracting one hundred 256×256 texture samples from each of the 30 Brodatz textures. The original texture images are 512×512 pixels, and the location of the smaller samples within the original is randomly selected. The total dataset consists of 3000 textures.

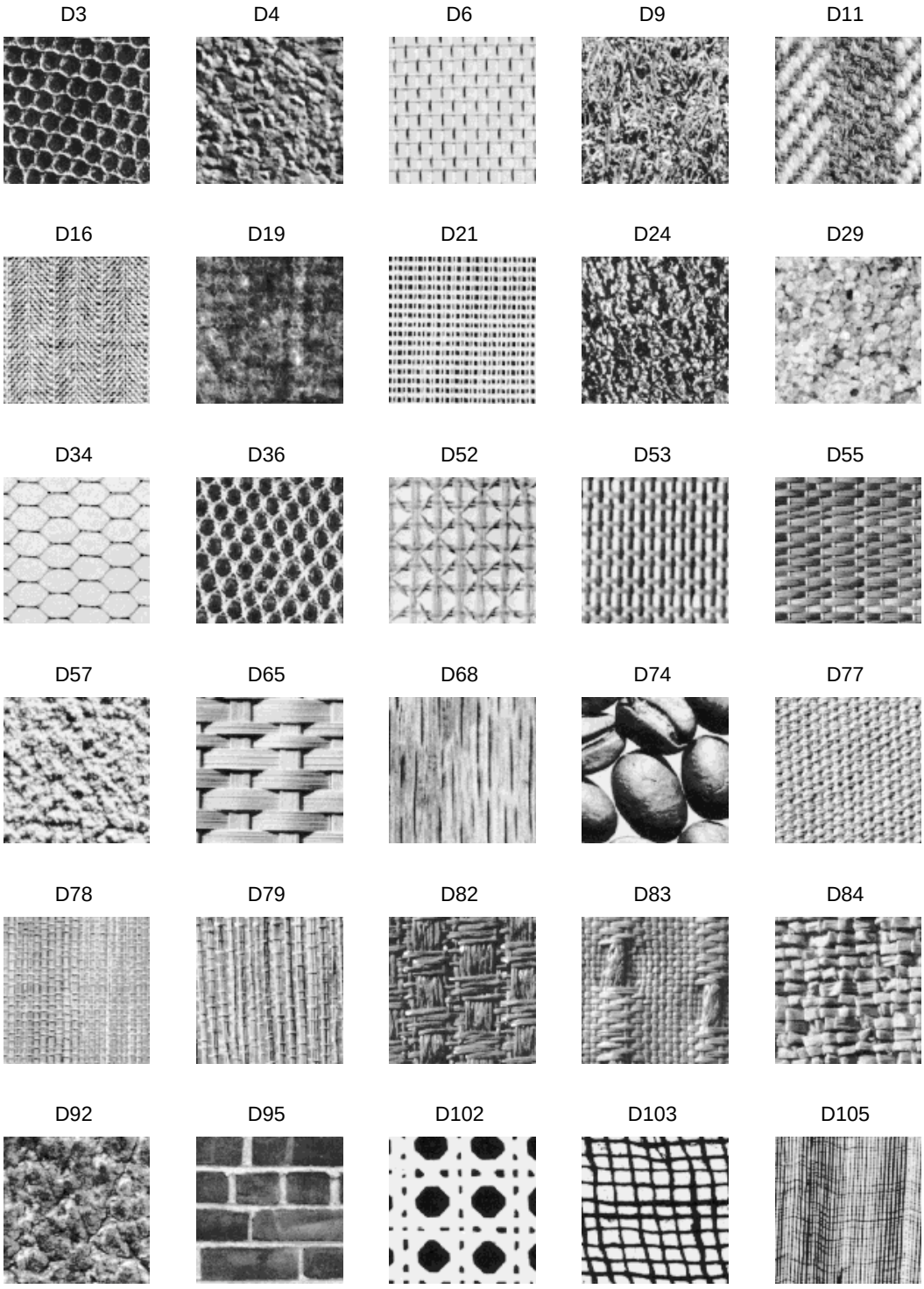


Figure 50: Texture set used in classification experiments

Table 2: Correct classification for 256 × 256 samples. System Config.: N=4, Parks-McClellan, $\omega_c = 0.5\pi$

L	4	6	8	10	12
$\mathbf{e}_1$	99.966	99.966	99.966	99.966	99.966
$\mathbf{e}_2$	99.966	99.9	99.9	99.9	99.9
$\mathbf{e}_3$	100	100	100	100	100

Table 3: Correct classification for 256 × 256 samples. System Config.: N=4, maximally flat, $\omega_c = 0.5\pi$

L	2	4	6	8	10	12
$\mathbf{e}_1$	98.933	99.966	100	99.966	99.966	99.966
$\mathbf{e}_2$	99.966	99.933	99.933	99.9	99.9	99.9
$\mathbf{e}_3$	99.8	99.933	100	100	100	100

Since there are parameter settings associated with the DFB (e.g., the lowpass filter cutoff ω_c, the number of bands N, the length of $\beta(z)$, L, and the design method for $\beta(z)$), we will have to evaluate the behavior of the system along these variables. Each combination of parameters will be referred to as a *system configuration*. The training and classification for each feature vectors was conducted using the *leave-one-out* algorithm as was done in [19].

The correct classification rates for different configurations are presented in Tables 2 - 7. Each table presents the classification results for different values of L in order to evaluate the impact of the directional filter responses. As L increases, the frequency selectivity of the wedge-shaped filters improves. Also, each row of a Table shows the performance for each of the energy measures e_1, e_2, and e_3.

The first thing to note is the excellent performance across all system configurations. All

Table 4: Correct classification for 256 × 256 samples. System Config.: N=8, Parks-McClellan, $\omega_c = 0.5\pi$

L	4	6	8	10	12
$\mathbf{e}_1$	100	100	100	100	100
$\mathbf{e}_2$	100	100	100	100	100
$\mathbf{e}_3$	100	100	100	100	100

Table 5: Correct classification for 256 × 256 samples. System Config.: N=8, maximally flat, $\omega_c = 0.5\pi$

L	2	4	6	8	10	12
$\mathbf{e}_1$	100	100	100	100	100	100
$\mathbf{e}_2$	100	100	100	100	100	100
$\mathbf{e}_3$	99.96	100	100	100	100	100

Table 6: Correct classification for 256 × 256 samples. System Config.: N=4, Parks-McClellan, $\omega_c = 0.25\pi$

L	4	6	8	10	12
$\mathbf{e}_1$	100	99.966	99.966	100	100
$\mathbf{e}_2$	100	99.966	99.966	99.966	100
$\mathbf{e}_3$	100	99.966	100	100	100

Table 7: Correct classification for 256 × 256 samples. System Config.: N=4, maximally flat, $\omega_c = 0.25\pi$

L	4	6	8	10	12
$\mathbf{e}_1$	99.733	99.555	99.366	99.433	99.533
$\mathbf{e}_2$	99.966	99.933	99.933	99.9	99.9
$\mathbf{e}_3$	99.933	99.933	99.9	99.966	100

of them consistently perform above 99.96% correct classification, with many configurations performing at the 100% level. Observations associated with these experiments may be summarized as follows:

- There is little variation in classification by increasing L, the length of $\beta(z)$. Even in a few instances, lower L performed slightly better, but the improvements were not significant. Hence, passband quality is not a relevant factor in this case.
- There is no performance advantage between a maximally-flat and Parks-McClellan design of $\beta(z)$.
- There is a marginal improvement when going from a 4-band configuration to an 8-band configuration. All, but one of the eight-band setups gave 100% correct classification.
- For the 4-band case, reducing ω_c from $\frac{\pi}{2}$ to $\frac{\pi}{4}$ gives inconclusive results. There is improvement in some cases while degradation in others. Since variations are not significant, we conclude that there is no advantage to exploit through the parameter ω_c.
- Similarly, we do not see a significant difference in performance among the energy features. We conclude that the results for e_3 on Table 2 are the best in terms of performance vs. computational requirements.

The DFB turns out to be a good tool for texture classification. The subband energy is separated well enough to allow for excellent feature discrimination power. We note that short length filters work as well as long ones. This situation seems to be a consequence of having good rejection of the filter stopbands rather than a flat response in the passbands. There is a significant difference in the quality of the passbands as L increases, while the stopband information is more consistent as L is varied. We seem to get more benefit from what is rejected as opposed to what is retained.

3.4.2 Impact of sample size

In the previous section we showed that the DFB provides excellent classification performance for the data set selected with texture samples of size 256×256. However, lower

Table 8: Classification performance for 64×64 samples. Best results as a function of L are in bold. System Config.: $N = 8$, maximally flat,$\omega_c = 0.5\pi$

L	4	6	8	10	12
$\mathbf{e}_1$	91.76	**92.07**	91.94	91.86	91.58
$\mathbf{e}_2$	84.71	**84.99**	84.88	84.56	83.89
$\mathbf{e}_3$	80.78	80.97	81.48	**81.66**	81.67

performance may be expected for smaller sample sizes. The smaller the sample, the less textural information can be extracted, i.e., there is less data to calculate estimates of $\mathbf{C}_k$ and $\mathbf{m}_k$. Additionally, if there are nonhomogeneous regions in the texture sample, then a smaller sample will make it difficult to average out the distortion caused by the defect.

To evaluate the impact of the sample size, the following variation on the experiments of the previous section was carried out. Each of the 512×512 textures was partitioned into overlapping 64×64 subimages with an overlap of 32 pixels among contiguous subimages. A total of 225 samples per texture were produced, for a grand total of 6750 texture samples.

We carried the classification experiment with $\omega_c = \frac{\pi}{2}$, $N = 8$, and maximally flat filter design. The classification of each feature vectors was conducted using the *leave-one-out* algorithm. The results are shown in Table 8. As suspected, the performance is severely affected by the sample size. Also, we note that as before, there is no advantage of using large values of L. On the other hand, we see that there is a significant difference in performance among the three energy measure. We find that the e_1 outperforms the other two energy measures. Slightly worse results were observed for the $N = 4$ case, and there was no performance difference for the two design methods.

To boost performance, we need to somehow increase the discriminating power provided by the feature set. One possible approach would be to form "super-vectors" by concatenating $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ in different combinations. We define the following feature vector combinations

$$\mathbf{f}_1 = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}, \quad \mathbf{f}_2 = \begin{bmatrix} \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix}, \quad \mathbf{f}_3 = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_3 \end{bmatrix}, \quad \mathbf{f}_4 = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix}. \tag{54}$$

Table 9: Classification performance for 64×64 samples. Best results as a function of L are in bold. System Config.: $N = 8$, maximally flat,$\omega_c = 0.5\pi$

L	4	6	8	10	12
$\mathbf{f}_1$	**96.68**	96.48	96.20	96.23	96.29
$\mathbf{f}_2$	**92.81**	92.37	92.19	92.02	91.64
$\mathbf{f}_3$	**95.77**	95.60	95.55	95.46	95.28
$\mathbf{f}_4$	**96.94**	96.59	96.29	96.53	96.11

The classification results for different combinations of feature vectors are shown in Table 9. The testing conditions are the same as those used for Table 8. We see a significant improvement in classification for all cases with $\mathbf{f}_1$ performing slightly better than $\mathbf{f}_2$ and $\mathbf{f}_3$, and performing equally to $\mathbf{f}_4$. Reasoning over the results of Table 9, it is not surprising we get these results since $\mathbf{f}_2$ consists of two features that are estimating the same quantity (i.e., variance). For the same reason, we can explain the similar performance between $\mathbf{f}_1$ and $\mathbf{f}_4$.

However, we are still short of the perfect classification achieved previously. Looking for further improvement, we tested the classifier with different values of ω_c. The conjecture is that we might see an improvement in classification as more highpass frequency information is retained. We used $\omega_c = \{\pi/2,\ \pi/4,\ \pi/8,\ 0\}$. The value of zero implies we have only removed the mean from the input image. The results of these experiments are discussed next.

We performed the experiments for the 64×64 sample set on all the configurations used for the 256×256 sample set. However, after further analysis of the data we report only a limited set of results based on the following observations:

- As discussed, we do not see any significant performance variation as a function of L. This is also true for $\mathbf{f}_i$, $i = 1, 2, 3, 4$. Hence, we only show the results at the endpoints $L = 4$ and $L = 12$.

- In general, we found $\mathbf{e}_1$ performs better than $\mathbf{e}_2$ and $\mathbf{e}_3$. For this reason, we only show results for $\mathbf{e}_1$.

- For a similar reason, we omit $\mathbf{f}_2$ and $\mathbf{f}_3$ and only show the classification rates for $\mathbf{f}_1$.

Table 10: Correct classification for 64×64 samples. System Config.: N=4, Parks-McClellan

feature vector	$\omega_c = \frac{\pi}{2}$		$\omega_c = \frac{\pi}{4}$		$\omega_c = 0$	
	L=4	L=12	L= 4	L=12	L=4	L=12
$\mathbf{e}_1$	87.54	86.6370	88.163	87.585	86.252	89.629
$\mathbf{f}_1$	94.993	94.385	97.926	97.126	97.541	97.541
$\mathbf{f}_4$	95.778	95.274	98.326	97.778	97.689	97.6

Table 11: Correct classification for 64×64 samples. System Config.: N=8, Parks-McClellan

feature vector	$\omega_c = \frac{\pi}{2}$		$\omega_c = \frac{\pi}{4}$		$\omega_c = 0$	
	L=4	L=12	L= 4	L=12	L=4	L=12
$\mathbf{e}_1$	91.793	91.052	92.627	93.319	98.429	98.519
$\mathbf{f}_1$	95.111	95.763	98.563	98.519	99.437	99.467
$\mathbf{f}_4$	95.896	95.674	98.429	98.844	99.289	99.333

- Since there is no significant performance difference with respect to using the Parks-McClellan or the maximally flat filter design methods, we present results only with the Parks-McClellan cases.

The classification results are shown in Tables 10 and 11. We see some improvement in classification as more frequency information is added. This is clearer for the cases with $N = 8$. The best system in terms of classification rate gives 99.47% correct classification rate with $\mathbf{f}_1$, $L = 12$, and $\omega_c = 0$.

The drawback to concatenating feature vectors is that the complexity of the classifier is increased significantly, both computationally and storage wise. In the worst case (for $\mathbf{f}_4$), we are required to invert, store, and do matrix multiplication with 24×24 covariance matrices. In section 3.5, we address the issue of feature reduction.

3.4.3 Selection of a set of reference system configurations

We have evaluated several DFB configurations and found that performance is very consistent across the degrees of freedom available for fine-tuning the system. Establishing a subset of these configurations based on performance would be difficult since there are groups of configurations that statistically could be considered equivalent as far as correct classification is concerned. Hence, we turn to other criteria to make this decision. We look at the

Table 12: "Best" DFB classifier configurations

Config. ID	Sample Size	Config. Parameters	Classif. %
[1mm] I	256×256	4 Band, $\omega_c = \frac{\pi}{4}$, $\mathbf{e}_1$, $L = 4$, PM	100
II	256×256	8 Band, $\omega_c = \frac{\pi}{2}$, $\mathbf{e}_1$, $L = 2$, MF	100
III	64×64	4 Band, $\omega_c = \frac{\pi}{4}$, $\mathbf{f}_4$, $L = 4$, MF	98.4
IV	64×64	8 Band, $\omega_c = 0$, $\mathbf{f}_1$, $L = 12$, PM	99.467
V	64×64	8 Band, $\omega_c = 0$, $\mathbf{f}_1$, $L = 4$, PM	99.437

computational complexity of the DFB, the energy feature, and the classifier dimensionality. The configurations selected are shown in Table 12. We note that configurations IV and V only differ on L, the length of the filters $\beta(z)$.

We have shown in this section that the DFB-based texture classification scheme developed to this point provides excellent performance with the great advantage that this DFB has a rather low computational complexity compared to other approaches reported in literature. In the following section we further simplify the complexity through the search of a minimal feature set.

3.5 Feature Reduction

In the previous section we have developed a texture classifier using the energy estimates of directional subbands as features. Our texture classifier uses the Bayes distance to do the class assignment. As discussed, the classifier is a supervised scheme that requires the computation of a mean vector and covariance matrix (plus the logarithm of its trace) for each class from a set of training patterns. Furthermore, testing entails the computation of the Bayes distance of each test vector to all texture classes. This involves multiplication of the test vector with the covariance matrices for each class, as can be seen in Equation (50). It is obvious that the dimension of the feature vector directly affects the computational complexity of the algorithm. Other classification schemes, parametric or non-parametric, supervised or unsupervised, show similar complexity dependency on pattern dimensionality.

Finding a minimal set of features can lead to significant resource savings. Moreover, tackling this issue, allows us to evaluate the significance of the features we have selected

and how relevant they are to our problem.

The brute force approach to feature reduction would require us to do a *full search* where every single combination of features is tested. For a set containing N_f features, this translates into

$$\sum_{k=1}^{N_f} \left(\begin{array}{c} N_f \\ k \end{array} \right),$$

different feature combinations which can become an untractable task for a moderate value of N_f. The *branch-and-bound* algorithm proposed by Narendra and Fukunaga provides a fast optimal feature selection algorithm. However it requires monotonicity of the classification function, which in real-life situations cannot be guaranteed.

Ranking features according to energy channel has been a popular approach for feature selection [19, 56]. This reasoning infers that channels that capture more energy will contain more information and they provide a better signature for the texture. However, this reasoning is flawed. For instance, there can be textures that exhibit similar structure and hence show similar energy concentration on certain subbands, then misclassification can occur between these two textures if the lowest energy channels are not taken into account. Additionally, there can be textures where a few subbands capture similar energy content, then energy ranking may not be consistent from one texture sample to the other.

Other feature selection approaches have been reported. Unser [128] uses a principal component analysis for texture segmentation. Mayer and Somol [85] have used genetic algorithms on texture-based SAR image analysis. A comprehensive evaluation of several feature reduction schemes (optimal and suboptimal) was reported by Jain and Zongker [54] on synthetic and real data. They concluded that the Sequential Forward Floating Selection (SFFS) algorithm from Pudil et al. [55] performs nearly as well as the optimal selection algorithm and is in general faster than the branch and bound algorithm.

In this thesis we use the SFFS for feature reduction. Pseudo-code is presented below (Algorithm 51). Before we go on, we consider the feature vectors $\mathbf{f}$ as a 1-D signal $f[j]$ for $j = 1, 2, ..N_f$. The ordering of features $f[j]$ is chosen according to the frequency plane partitions of Figure 2. Hence $f[1]$ corresponds to the energy of the directional subband

1: Start with $\mathbf{f}_0 = \emptyset$, $k = 0$, $j = 0$ and the full K-dimensional feature set.
2: Step 1. (Inclusion)
3: **while** j ¡= K **do**
4: Select best feature
5: $f^+ = \arg\max_{f \epsilon \mathbf{f}_k} J(\mathbf{f}_k - f)$
6: $\mathbf{f}_{k+1} = \mathbf{f}_k + f^+$; $k = k + 1$; j = j+1
7: **end while**
8: Step 2. (Conditional Exclusion)
9: $f^- = \arg\max_{f \epsilon \mathbf{f}_k} J(\mathbf{f}_k - f)$
10: Step 3.
11: **if** $J(\mathbf{f}_k - f^-) > J(\mathbf{f}_k)$ **then**
12: $\mathbf{f}_{k+1} = \mathbf{f}_k - f^-$
13: goto Step 3
14: **else**
15: goto Step 2.
16: **end if**

Figure 51: Pseudo-code for the Sequential Floating Forward Selection (SFFS) algorithm

Table 13: Incremental classification for system configurations I and II using SFFS. j is the feature index on the original feature vector $\mathbf{f}$

Num. of features	SFFS evolution			
	I		II	
	j	% correct	j	% correct
1	1	62.13	5	70.33
2	3	99.20	2	98.67
3	2	100.0	7	100.0

1, $f[2]$ corresponds to the energy of subband 2, etc. The algorithm starts with a set of I features $\mathbf{f} = \{f_1, f_2, ..., f_I\}$ that are used to find the subset $\mathbf{f}_{best}$ by evaluating subsets of features with an objective function $J(\cdot)$. In our case the objective function value is the correct classification rate of the texture classifier.

We applied the SFFS algorithm to each one of the configurations from Table 12. For configurations I and II, we present the evolution of the SFFS algorithm in table 13. In this case, we see that both system configurations I and II achieve 100% classification with only 3 features. System I would be the preferred system as it only requires calculation of a four-band DFB, and has better performance than system II for the case when only the best two features are used.

The remaining three systems did not achieve 100% classification, so it is possible for SFFS to find a feature subset that improves classification.

Table 14: Incremental classification for system configuration III using SFFS. ... j is the feature index on the original feature vector **f**

feature count	j	%
1	4	33.39
2	6	69.29
3	7	86.31
4	3	94.60
5	2	96.78
6	1	97.79
7	11	98.41
8	8	98.64
9	9	98.79

For system III, SFFS found a minimal set of 9 features out of 12. The incremental classification improvement is shown in table 14. The classification rate from table 12 is slightly improved from 98.4% to 98.79%. Nonetheless, what is important in this case is the reduction of the feature set by three features, which translates into lower complexity. Another perspective to this evaluation is that we can achieve 98.4% correct classification using only seven features.

In this and the remaining systems, adding more features beyond the minimal set decreases the classifier performance. This is the so-called *curse-of-dimensionality* problem.

Similar observations can be made for system configurations IV and V. For system IV, we see in Table 15 a small improvement from 99.47% to 99.61% using only 11 out of 16 features. To maintain classification rates above the 99% level, it is only necessary to use eight features. For system V, we show in Table 16 that with only 10 features we reach 99.62% correct classification. This last result is the best classification performance obtained with the DFB-based systems evaluated in this thesis, and translates into 26 misclassified texture samples out of a set of 6750.

3.6 Comparison with other classification schemes

In this section we compare the performance of the DFB-based classifier to other multichannel schemes considered as state of the art. We show that the DFB compares favorably against other multichannel schemes. Our comparison refers to the work of Chang and Kuo [19].

Table 15: Incremental classification for system configuration IV using SFFS. ... j is the feature index on the original feature vector $\mathbf{f}$

feature count	j	%
1	11	29.68
2	6	67.26
3	13	88.05
4	8	94.10
5	4	96.40
6	7	97.76
7	1	98.58
8	15	99.14
9	2	99.23
10	10	99.53
11	9	99.61

Table 16: Incremental classification for system configuration V using SFFS. ... j is the feature index on the original feature vector $\mathbf{f}$

feature count	j	%
1	5	23.98
2	4	59.44
3	6	84.99
4	7	93.95
5	13	96.14
6	10	97.64
7	8	98.47
8	15	98.64
9	16	99.09
10	3	99.62

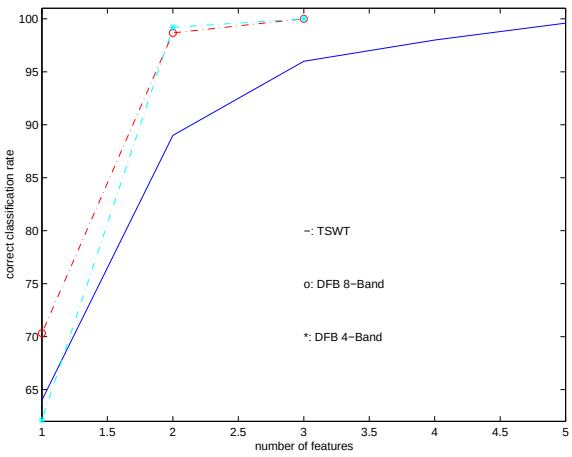


Figure 52: Comparison of DFB-based classification with the TSWT as the number textures is increased.

They have reported a comparison among different schemes.

First, we compare the results between the TSWT and DFB using the 256×256 samples. The TSWT was calculated using the Battle-Lemarié cubic spline wavelet truncated to 16 coefficients. The TSWT feature vector size was varied from 1 to 5, and the features were selected according to the energy ranking criteria. The results are plotted on Figure 52 together with the results from the DFB configurations I and II from Table 12.

We see the improvement in performance on a feature by feature case. The TSWT achieves only 99.6% correct classification for five features, while the DFB-based systems give 100% correct classification with only *three* features. Another significant result is that with only two features the DFB achieves classification rates that surpasses the TSWT by about 10%.

For the of 64×64 texture samples Chang and Kuo compared the 2-D DWT, Gabor functions, 5×5 DST, 5 DCT, 4×4 DHT, and Laws filters. The last four schemes are the local transforms used by Unser in [126], and they all have almost identical performance. Hence, only the DST results are discussed. Except for the image decomposition, all other experimental settings are the same. Since the results from [19] are only reported graphically, we have derived them by careful direct measurement over the published graphs.

We use DFB configurations IV and V described in Table 12. The DFB performance is

similar to the TSWT and DST which are the top performers in [19]. We plot the performance as a function of the number of features in Figure 53. For the DFB we use the feature set derived with the SFFS algorithm.

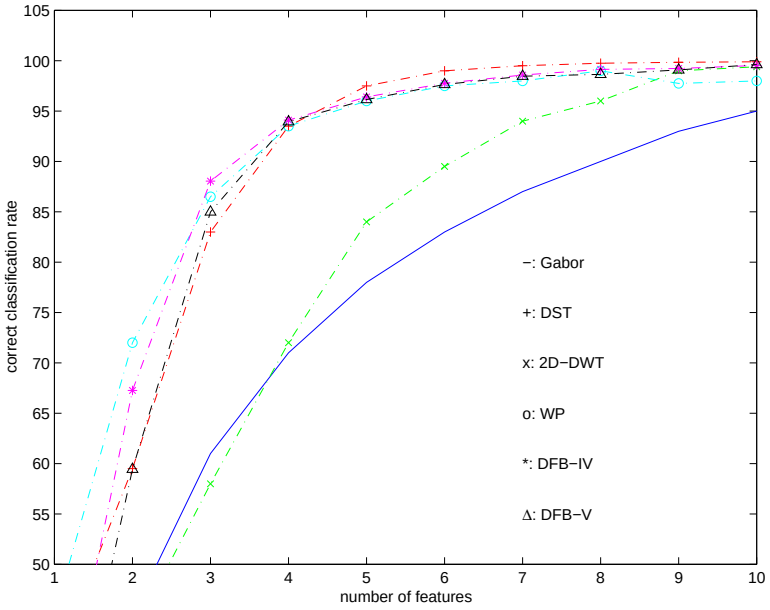
In particular, the local transforms present good performance with five or more features, while the TSWT seems to degrade for the same cases. The DFB performance curves seem to be more consistent as the number of feature increases, and is slightly below the DST at the 10 feature mark. We can conclude that the DFB is second in to the DST purely based on classification rate.

The good DFB performance seems to be related to a couple of factors. First, greater directional resolution provides better results than other transforms like the TSWT and 2-D DWT which are limited to three orientations. Second, the quality of the filter responses seems to have a significant role too. As seen in Figure 53, the 2-D Gabor decomposition was the worst performing system even though it has similar directional selectivity to the DFB. This behavior points towards issues on the quality of the Gabor function passbands. Since the discrete coefficients of Gabor filters are obtained by direct sampling of the continuous functions, then there is a tradeoff between sampling density and aliasing. Finally, as 2-D Gabor filters are non-separable functions, the computational advantage of the DFB over Gabor increases the appeal of our scheme.

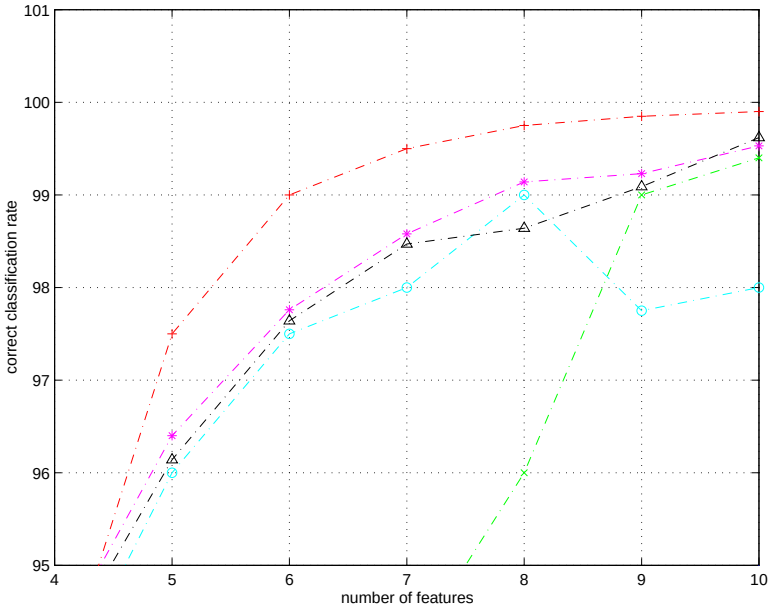
3.7 Chapter Summary

This chapter consisted of an extensive study on the development of a multichannel texture classification algorithm using the ladder-based biorthogonal DFBs introduced in this thesis. We have adopted a set of energy based features and used Bayes distance as a classifier under certain assumptions of feature distribution and independence. The good classification performance indicates that the empirical data is well modelled by these assumptions. We have also analyzed the significance of features, and produced a reduced feature set for each of the test configurations using the SFFS algorithm.

We have shown top performance when compared to other multichannel schemes considered state of the art, like the DST and other local transforms. In particular, as texture



(a) Correct classification using Gabor functions, DST, DWT, TSWT, and DFB.



(b) Closer view of classification plot.

Figure 53: Comparison of DFB-based classification with other multichannel schemes.

sample size increases, we see a clear advantage of the DFB as the computational requirements are combined with performance requirements. The lower computational complexity does not detract the DFB from achieving good orientation selectivity and good classification results.

An inherent property of the DFB is steerability, i.e., the possibility of analyzing 2-D signals with respect to rotation. in the next chapter, we explore the use of this property to design a texture classifier that is invariant to rotations.

CHAPTER IV

FURTHER APPLICATIONS OF THE DIRECTIONAL FILTER BANK TO TEXTURE

In the previous chapter, we made a detailed experimental analysis of the DFB on texture classification. We found that the features derived from the DFB provide excellent discriminability, which gives some of the best classification results to date. In this chapter, we approach three more problems using the DFB as an analysis tool. First, we revisit the texture classification problem for the case of rotation invariant classification. Next, we look at texture segmentation using a well known multichannel system, and compare it against several other multichannel decompositions. Finally, we evaluate the DFB as part of an image pyramid on texture synthesis using the Heeger-Bergen algorithm.

4.1 Rotation Invariant Texture Classification

In the real world, a computer vision or image analysis system will encounter instances of textures that have some type of distortion with respect to the samples used for system training. These distortions can be caused by changes in illumination, scale, perspective, skew and rotation. Our visual system catches these variations easily, but it is a non-trivial problem in image analysis research.

The objective of this section is to implement a system capable of identifying textures independent of rotation. The orientation selectivity of the DFB allows us to produce a set of rotation invariant (RI) features.

The problem of RI texture classification has been addressed in a few works. One of the early approaches was the use of circular AR models [68]. Cohen, et al. [21] used GMRF as models. The model parameters were used as features for the classifier. The combination of GMRFs with wavelets was proposed by Porter and Canagarajah in [101].

Multichannel approaches are particularly well suited to deal with rotation invariance.

Leung and Peterson [74] used a Gabor representation to identify a rotation angle with respect to a reference image. You and Cohen [140] used filters specifically tuned to a training set. Greenspan et al. [44] used a steerable pyramid and DFT encoding to obtain RI features. Haley and Manjunath [45, 46] used a 2-D analytical polar Gabor representation, and formed a set of features based on the frequency magnitude, phase, and autocorrelations of the features.

Hill, et al. [51] used the DT-CWT to generate isotropic and anisotropic RI features including a DFT-encoding step. Vautrot et al. [132] used 2-D continuous wavelet theory to design non-separable oriented wavelet filters. Other approaches [20, 137] have model subband dependencies using Hidden Markov Models (HMMs). A recent and novel approach was proposed by Ojala, et al. [91] that uses grayscale invariant local circular feature distributions.

4.1.1 A DFB-based rotation invariant feature set

In this section, we show that the DFT-encoding approach used in [44, 46, 51, 132] is also applicable to the feature sets derived from the DFB.

Define the 2-D rotation operator as $\mathcal{R}_\theta(\cdot)$ which performs of a rigid rotation by an angle θ. The rotation of the axes is implemented as

$$\begin{bmatrix} t_0' \\ t_1' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} t_0 \\ t_1 \end{bmatrix} \tag{55}$$

We start by letting $x(t_0, t_1)$ be a continuous time texture. If the texture is rotated then we get

$$x'(t_0, t_1) = \mathcal{R}_\theta(x(t_0, t_1))$$

for some θ. Then, if the Fourier transform of $x(t_0, t_1)$ is given by

$$X(\Omega_0, \Omega_1) = \mathcal{F}(x(t_0, t_1))$$

and likewise

$$X'(\Omega_0, \Omega_1) = \mathcal{F}(x'(t_0, t_1))$$

it can be shown that

$$X'(\Omega_0, \Omega_1) = \mathcal{R}_\theta(X(\Omega_0, \Omega_1)).$$

In other words, a rotation in the (continuous) spatial domain generates a similar rotation in the (continuous) frequency domain. The proof is easy by simple variable substitution of Equation (55) in the spatial and frequency variables.

When the images are acquired and digitized, then the above relationships do not hold in general. However, we assume for practical purposes that

$$x'[n_0, n_1] \approx \mathcal{R}_\theta(x[n_0, n_1]),$$

and

$$X'[\omega_0, \omega_1] \approx \mathcal{R}_\theta(X(\omega_0, \omega_1)).$$

On the other hand, the frequency plane partitioning generated by the DFB is fixed. Therefore, as the texture is rotated, the frequency content is redistributed to different subbands. How the texture energy is redistributed will depend on both the angle and the number of bands in the DFB. We would expect that the performance of the classifier from the previous chapter will degrade when a texture sample has gone through a large rotation with respect to the training set.

A way to generate RI features is to do the so-called "DFT encoding." Suppose that we use a DFB with a large number of subbands N. (In the limit, a subband will become a diametral slice of the frequency plane.) Hence, we have an N-dimensional feature vector

$$\mathbf{f} = [\ e_0 \quad e_1 \quad e_1 \quad \ldots \quad e_{N-1}\ ]^T,$$

where e_i is an energy measure of the i^{th} subband. We can map $\mathbf{f}$ to a discrete 1-D signal $f[n]$ for $n = 0, 1, \ldots, N-1$.

Next, we define the circular shift operator for $f[n]$ as $S_\ell(f) = f((n-\ell) \mathrm{mod} N)$. Then we can establish the following relationship. Suppose we have a texture $x[n_0, n_1]$ with a feature vector described by the 1-D sequence $f[n]$.

$$\mathcal{R}_\theta(x[n_0, n_1]) \Rightarrow S_\ell(f[n]). \tag{56}$$

In words, the rotation of a texture implies a circular shift of the feature vector.

In reality, the circular shift on the feature vector is not exact since we are working with a discrete sequence of energy measures while θ is a continuous variable. As N gets smaller (e.g., the case of eight- and four-band DFBs), the rotation-to-shift relationship gets coarser and could become inaccurate.

This rotation-to-shift relationship allows us to easily appreciate how an RI feature set can be obtained. Since the difference between a reference and rotated texture is only a circular shift on the feature vectors, then the shift can be eliminated by adding a *DFT encoding* step.

This scheme consists of first taking the N-point DFT $F[k]$ of the feature sequence $f[n]$. Next, the magnitude coefficients $|F[k]|$ are taken to get RI features. The RI feature vector is given by

$$\mathbf{F} = [\ |F[0]| \quad |F[1]| \quad \dots \quad |F[N/2+1]|\ ]^T. \tag{57}$$

Since the DFT magnitude is an even function in this case, the new feature vectors have only $\frac{N}{2}+1$ elements. Therefore RI comes at the cost of reducing the feature space. On one hand, this has the positive effect of reducing computational complexity. On the other hand, if N is small, we could lose significant classification performance. We discuss these effects in the subsequent subsections.

4.1.2 Data set and reference work for RI classification

We use the same data set tested by Haley and Manjunath in [46]. This image set is available over the internet at [119], and consists of 13 textures (mostly from the Brodatz album). Each texture is scanned and digitized at rotations of 0^o, 30^o, 60^o, 90^o, 120^o, and 150^o. Each image is 512×512. Each image is then subdivided into 16 128×128 images which are used for training or testing. Samples of the data set are presented in Figure 54.

Haley and Manjunath use a polar, complex valued 2-D Gabor representation consisting of four resolution levels and 16 orientations. The increase in data is 64-fold. For a texture $T(i,j)$, a feature set is calculated for each spatial location (i,j) from the 64 channels. These "microfeatures" collect information about the amplitude, the frequency and the orientation

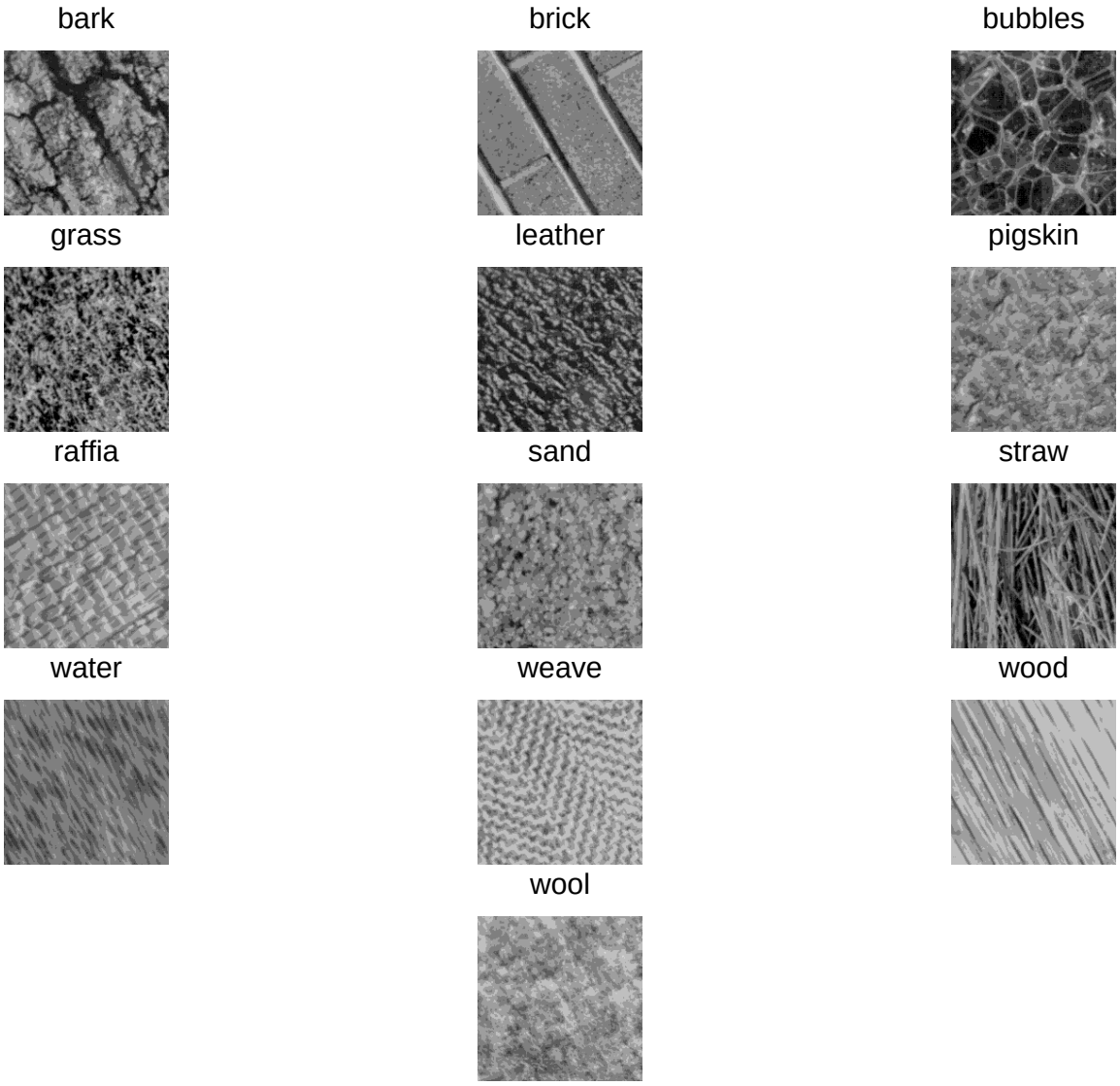


Figure 54: Texture set used in RI classification experiments, images are rotated by 30^0

Table 17: DFB Rotation invariant classification results as a function of ω_c.

ω_c	0	$\pi/16$	$\pi/8$	$\pi/4$
N=4	50.64	66.11	66.99	61.62
N=8	50.32	72.04	72.12	68.35

of the image at that specific location. Macrofeatures or global features are obtained from the microfeatures, and are used as a feature set for a Bayesian classifier. Such an elaborate model also creates high dimensionality of the feature space with a 204-dimensional feature vector. Haley and Manjunath got a correct classification rate of 96.4% on the data set described above.

4.1.3 RI classification experiments using the DFB

We start with a similar system set-up to the one used in Chapter 3. As before, ω_c is the cutoff frequency of the lowpass filter. The highpass component is processed with the DFB. In this case $\beta(z)$ was designed with the Parks-McClellan program using $L = 12$. As a feature, we use the $\mathbf{e}_1$ energy measure from Equation (46). At the feature extraction stage, we additionally do the DFT encoding step. We then use the leave-one-out method to train and test the classifier.

Our first results are shown in Table 17. The best result for this system is 72.12%. There are a few possible reasons for this poor performance. First, as in the previous chapter, the image size might not be large enough to capture the characteristics of the texture. Second, the number of features is reduced to three for the four-band DFB, and five for the eight-band DFB. Finally, as stated previously, the degree of directional selectivity might be too coarse, so the DFT encoding step does not capture accurately the rotation-shift relationship from Equation (56).

To improve classification, we increase the number of features by concatenating different energy measures as in Chapter 3. We use the super-vector $\mathbf{f}_4$ from Equation (54). The DFT encoding step is applied to each energy vector $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$ separately, before forming the super-vector. We show the result for an eight-band DFB in Table 18. We see that this approach improves classification; in this case, we achieved 91.35% for $\omega_c = \pi/8$.

However, there is still room for improvement. Since the DFB is a maximally-decimated

Table 18: DFB Rotation invariant classification results as a function of ω_c using a combination of three energy features.

ω_c	0	$\pi/16$	$\pi/8$	$\pi/4$
N=8	68.67	90.14	91.35	87.66

system, increasing N decreases the subband size, making the energy estimates unreliable. Using the UDFB (section 2.4) would avoid this issue, but this comes at the expense of an N-fold storage and computational increase.

As we have discussed, the DFB does not differentiate between coarse or fine information along a particular direction. In Chapter 2, we introduced a family of directional pyramids (DPs) to address this issue. We select the DP structure from Figure 46-(a). In this case, the image is decomposed into J undecimated pyramid levels. Except for the lowpass channel, each of the J-1 bandpass channels is decomposed by the maximally decimated DFB. Hence, we have a J-fold increase in data.

The subband energies for each pyramid level are grouped to form a feature vector $\mathbf{f}^{(j)}$ for $j = 1, 2, ..., J-1$, and then are DFT encoded to form RI vectors $\mathbf{F}^{(j)}$. The feature vectors for all levels are staggered on a single feature vector given by

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}^{(1)} \\ \cdots \\ \mathbf{F}^{(2)} \\ \cdots \\ \vdots \\ \cdots \\ \mathbf{F}^{(J-1)} \end{bmatrix}. \tag{58}$$

We note that the lowest frequency channel is not used. The feature vector has dimension $(J-1) \times (\frac{N}{2} + 1)$.

In Tables 19 and 20 we show classification results using the $\mathbf{e}_1$ energy [Equation (46)] as a feature. We show the performance progression as J is increased. Beyond $J = 5$ there was no improvement in classification. The tables show the benefit of decomposing image information at different resolutions in addition to an angular decomposition. We draw the

Table 19: Directional Pyramid Rotation invariant classification results for $N = 4$

	J=2	J=3	J=4	J=5
L=4	49.60	78.13	91.59	94.95
L=12	49.28	84.05	95.35	97.28
L=18	49.52	84.13	95.19	97.04

Table 20: Directional Pyramid Rotation invariant classification results for $N = 8$

	J=2	J=3	J=4	J=5
L=4	47.92	81.81	92.95	93.99
L=12	55.93	92.39	96.88	96.71
L=18	54.81	92.16	96.96	96.88

following observations from these tables.

- We observe an improvement as a function of the ladder filter length L, from $L = 4$ to $L = 12$, however for values greater than 12, increasing the length of the filters does not show a significant benefit.

- With just two resolution levels, classification jumps from the 54%-56% range to more than 92% correct classification.

- The best classification result is 97.28% given by the system with $J = 5$ and $N = 4$. Hence, in this case, we see that higher directional selectivity does not necessarily improve classification.

- For the $N = 8$ systems, we seem to reach a plateau on classification for $J \geq 4$, which implies a 15-dimensional feature vector. The best performance is at 96.96% correct classification.

Using maximally-flat design for the ladder filters did not improve performance of the classification. The energy measures given by Equations (47) and (48) show a significant loss of performance. For instance, using $\mathbf{e}_2$ with $J = 5$, $N = 4$, and $L = 18$ gives 91.83% correct classification.

Compared to the work of Haley and Manjunath [46], we achieve slightly better performance with only 12 features per texture sample, while they require 204 features.

4.1.4 Section discussion

We have shown that the DFB-based DP provides a set of RI features with high discriminative power over the data set. Compared with the work of Haley and Manjunath we achieve the same performance at a significantly lower computational and storage cost with a very simple feature set. This implies that a complete model and feature set is unnecessary for RI multichannel classification.

As future work, a more extensive study of the RI properties is desirable. There is still the need to compare against other multichannel representations. Under more general "real life" conditions, a rotation could be presented along some other type of geometrical transformation, i.e., skewings, non-linear deformations, etc. An open problem for future work is the development of texture classification systems invariant other types of geometrical distortions.

4.2 DFB-based Multichannel Texture Segmentation

As presented in the previous chapter, texture segmentation has been an active area of research for a few decades. Ultimately, it is sought to understand the mechanisms that allow our visual system to easily understand a scene according to its textural characteristics. In signal processing terms, the image received by the eye is decomposed into a dyadic representation whose partitioning of the 2-D frequency plane is similar to what is shown in Figure 47-(b). We note that our visual system is capable of a much higher degree of angular and radial resolution.

Multichannel texture segmentation was pioneered by Laws in [72] using 25 simple 2-D separable kernels. Subsequently, a lot of attention was placed on Gabor functions because the similarity of space-frequency characteristics with those measured experimentally from mammals. In particular the work of Jain and Farrokhnia [56] has explored this relationship in detail from an engineering perspective. A drawback with Gabor functions is that the implementation is computationally expensive because the 2-D discretized filters are large and non-separable. To deal with complexity, other image decompositions have been used as the front end of the segmentation system. For instance, work with the DCT [106], wavelet

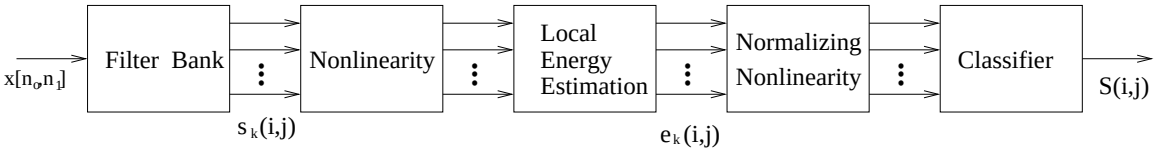


Figure 55: Classical segmentation system based on multichannel filtering

transforms [102, 75], wavelet packets, wavelet frames [128], and complex wavelet transforms [69] has been reported with different degrees of success.

Randen and Husøy [106] did an extensive evaluation of texture segmentation algorithms with a heavy emphasis in multichannel systems. Their work allows a fair ranking and characterization of the major approaches reported to date. In subsequent sections we adhere to their framework to evaluate and rank DFB-based systems.

A block diagram showing a typical multichannel segmentation system is presented in Figure 55. In general, the multichannel system can be split into three stages: filter bank stage, feature extraction stage, and classifier stage. The first stage as we have seen decomposes an image on a set of channels that extracts characteristics of the input image at different scales and orientations.

The contribution of this section is to evaluate the DFB as an alternate front end to the "classical" multichannel segmentation system, and evaluate its performance against other filter banks. However, it is necessary to consider some significant design issues to adapt this scheme to the DFB; these will be described in subsequent subsections.

4.2.1 Feature Extraction Stage

Intuition suggests that the textures composing the image will respond differently to each filter. For instance, a structural texture will have most of its energy concentrated at certain channels, while stochastic textures will have their energy more evenly distributed across bands. Measuring the local interaction of the subband coefficients allows us to characterize the overall properties of the texture.

The feature extraction stage is a set of pre-conditioning operations whose objective is to introduce some degree of smoothing or homogeneity on each channel $s_k[i, j]$ such that texture regions approach uniformly shaded blobs easy to recognize.

The feature extraction stage consists of the second, third, and fourth operations depicted in Figure 55. First, each channel is passed through a non-linearity in order to rectify the oscillatory nature of the channels. As discussed in [56], the non-linearity acts as a "blob detector" which identifies primitive shapes and structures. Jain and Farrokhnia used

$$f(x) = \tanh(\alpha x). \tag{59}$$

which has a sigmoidal shape. A value of $\alpha = 0.25$ has been widely used. The absolute value $f(x) = |x|$, and the squaring function $f(x) = x^2$, have been popular non-linearity choices too [106, 129].

Ideally, we would like to extract the primitives and placement rules for each texture in the image and use them to identify the textures. However, this is a rather difficult task. In order to obtain some information about the interactions among these blobs, a measure of local energy is used. This operation consists of doing spatial smoothing with 2-D filters $g_k[i, j]$. The energy map for the kth channel is given by the convolution

$$m_k[i, j] = g_k[i, j] * f\{s_k[i, j]\} \tag{60}$$

Intuitively, averaging over a region with statistically similar primitives would produce a slow varying response that would be different from other textures within the image.

The size and response of the $g_k(i, j)$ functions should be carefully selected. There are different opposing constraints that should be kept in mind:

- We want the filter dimensions to be as large as possible to have a good statistical representation of the primitives.
- Very large filters will preclude us having good localization around the region boundaries, where we will effectively be averaging blobs from different textures. Hence, filter support should be as small as possible.
- Channels at coarser resolutions produce larger blobs which require large window size.
- High frequency channels capture finer detail requiring small averaging windows.

Gaussian windows have shown to be a good compromise among this set of requirements. They are known to provide optimal time-frequency localization. The basic 1-D Gaussian response is given by

$$g[n] = \frac{1}{\sqrt{2\pi}\sigma_s} exp\{-\frac{1}{2}\frac{n^2}{\sigma_s^2}\} \tag{61}$$

where the σ_s parameter is specified as a function of the subband frequency content. As defined in [56], this parameter is dependent on the average channel frequency u_0 (i.e. the centroid), and it is given by

$$\sigma_s = \frac{1}{2\sqrt{2}u_0} \tag{62}$$

The size of the window is set to $2\sigma_s$. It is a straightforward step to derive the 2-D filters $g_k[i,j]$.

The *normalizing* non-linearity is used to prevent large dynamic range variations amongst the energy maps. Non-normalized energy maps could lead to feature vectors where one or only a few features dominate. This stage consists of some type of normalization, e.g., scaling to the $[0,1]$ range, scaling to unit variance, etc.

4.2.2 Classification Stage

As the last step of this stage, the energy maps obtained from each channel are combined to form feature vectors. In segmentation, each pixel on the original image must be assigned to some class. For a filter bank with N channels, each image pixel will count with an N-dimensional feature vector $\mathbf{f}_{i,j}$ given by

$$\mathbf{f}_{i,j} = [m_1(i,j) \;\; m_2(i,j) \;\; ... \;\; m_N(i,j)]^T, \tag{63}$$

where $m_k(i,j)$ corresponds to the 2-D energy map of the k^{th} channel. The $\{i,j\}$ indices refer to a spatial location in both the original image and the energy map.

The feature vectors $\mathbf{f}_{i,j}$ are passed through a classifier to be assigned to one of the C texture classes in the image. In this thesis we assume that C is known beforehand. The final output of the classifier will be a classification map $S(i,j)$ which assigns each pixel in the original image to one of the textures classes.

Many different types of classification schemes considered. Supervised and unsupervised algorithms, neural networks, genetic algorithms, etc. have been studied. Clustering algorithms have been a popular choice. Assuming that the ground truth segmentation maps are available, we use classification error as a measure of accuracy.

4.2.3 Framework for Texture Segmentation with the DFB

In a previous work [111] we introduced a segmentation system based on the DFB. The results showed that the DFB is well suited for segmentation. In this case we used the non-linearity from Equation (59), and the fuzzy C-means clustering algorithm for classification.

Although this system performs well, in order to compare against other systems we would have to build a similar system for other multichannel representations. However, it turns out this type of comparison has already been reported by Randen and Husøy in [106]. They developed a framework which allows testing of many multichannel representations while keeping the other elements of the system in Figure 55 the same. Therefore, a fair characterization of the filter banks can be made. The decompositions tested by Randen and Husøy are presented in Table 29 in subsection 4.2.7.

Their work stands out because they have made their programs and data sets easily accessible on the internet [105]. Obviously, the great benefit of this is that new multichannel representations have a point of reference against which their segmentation performance can be evaluated. As an example, De Rivaz has reported results for the DT-CWT in [109]. Hence, for comparison purposes, it makes sense that we adopt this framework.

The image data set for this framework consists of the 12 texture mixtures shown in Figure 56 and Figure 57. The mixtures exhibit different degrees of difficulty in terms of number of textures and boundaries. In some cases, textures are difficult to identify by a human observer. Training data corresponding to each texture mixture and "ground truth" maps are also made available.

Texture data had their histograms equalized, so that textures could not be discriminated based on their first order statistics. Finally, to normalize the dynamic range of the feature vectors for a given filter bank, the feature vectors were scaled by the normalization factors

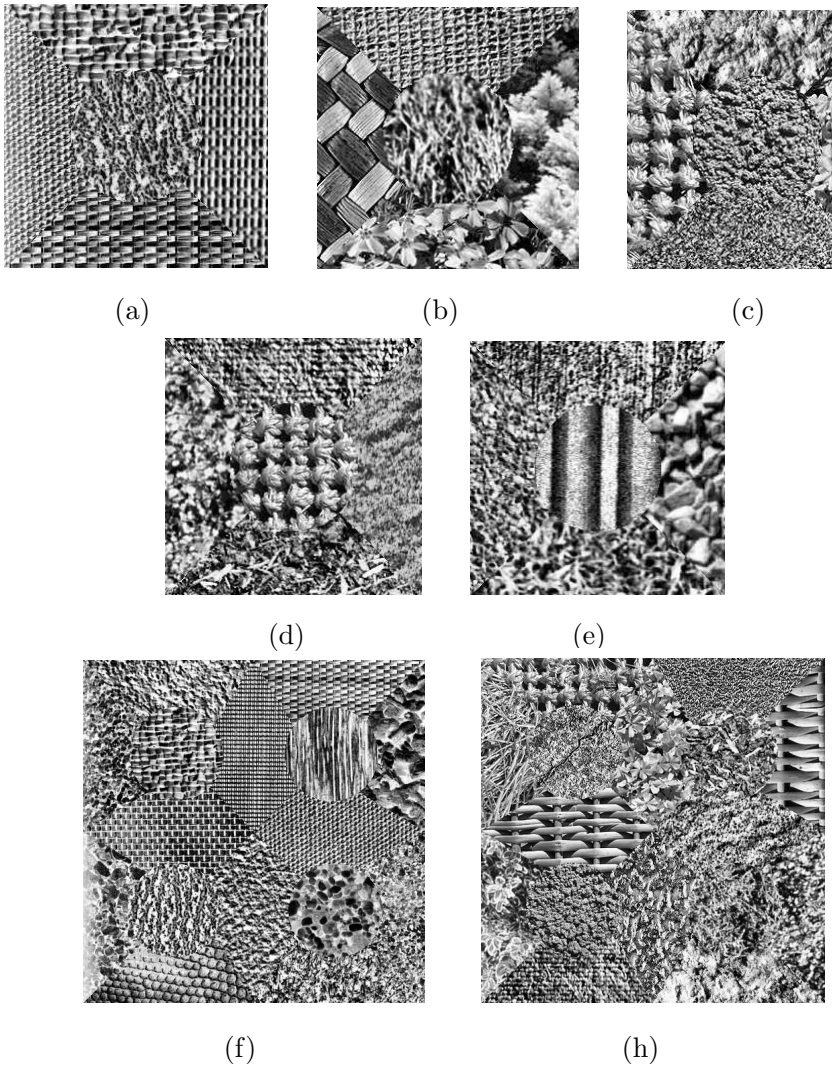


Figure 56: Test texture mixtures from [106]

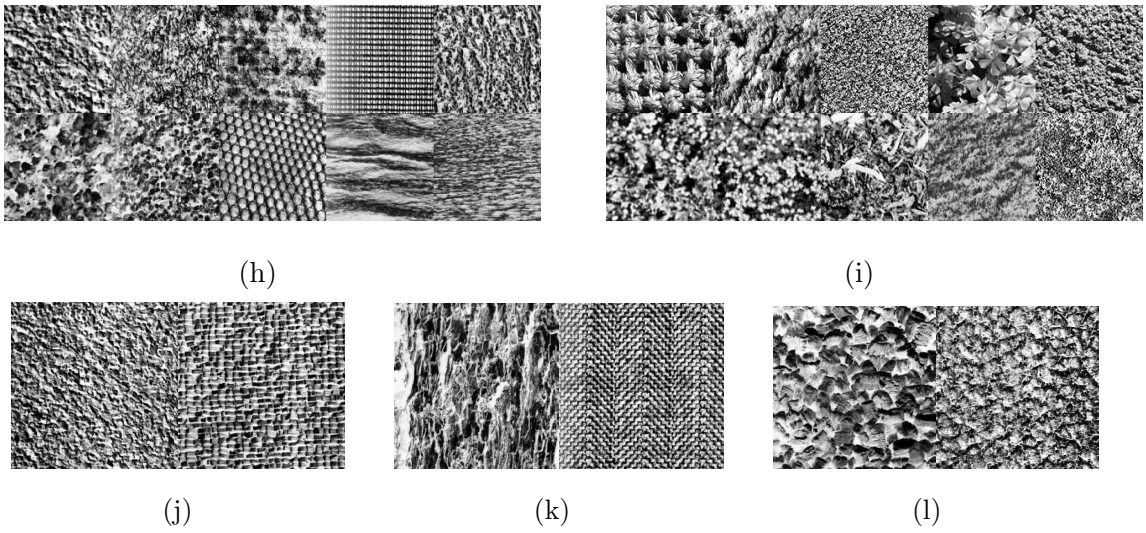


Figure 57: Test texture mixtures from Randen and Husøy [106]

which give unit variance to the features of the mixture in Figure 56(a).

Besides the multichannel representation, the remaining blocks in the Husøy-Randen segmentation system (Figure 55) are implemented as follows:

- On the feature extraction stage, the rectifying non-linearity is given by

$$f_1(x) = |x|^2, \tag{64}$$

  the normalization non-linearity is also used, and is given by

$$f_2(x) = \log(x). \tag{65}$$

  This combination of non-linearities was found to give the best segmentation results in a work by Unser and Eden [129].

- A 2-D Gaussian kernel was used to estimate the local energy as described in subsection 4.2.1.

- The Learning Vector Quantization (LVQ) algorithm from Kohonen [70] was selected as the classifier. This is a supervised scheme requiring training and testing data. Source and executable code is available at [79]. More specifically the `olvq1` program

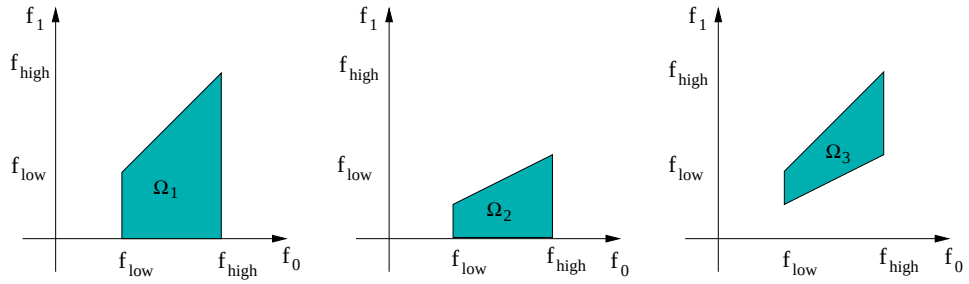


Figure 58: Basic frequency domain supports for DFB truncated subbands. Ω_1 is for the four-band DFB case, Ω_2 and Ω_3 are for the eight-band DFB case.

was used, which automatically selects some classifier parameters based on the data. A codebook size of 800 vectors was selected for the framework.

For the Gaussian kernel $g[i,j]$ it is necessary to estimate the filter parameter σ_s in Equation (61). For Gabor functions σ_s is proportional to u_0, the center frequency. In the DFB case, u_0 can be viewed as the "average" frequency of a subband. The directional subbands have a truncated wedge-shaped support as shown in Figure 58. We can identify three basic subband shapes. We have Ω_1 for the four-band DFB, and Ω_2 and Ω_3 for the eight-band case. Hence, the centroids of the regions represent the average coordinates $(\overline{f}_0, \overline{f}_1)$ over a 2-D region. From calculus we have

$$\overline{f}_i = \frac{1}{Area(\Omega)} \int \int_{\Omega} f_i df_0 df_1$$

for $i = 0, 1$. Evaluating the integrals over Ω_1 we get

$$\overline{f}_0 = \frac{1}{3A_1}(f_h^3 - f_\ell^3)$$

$$\overline{f}_1 = \frac{1}{6A_1}(f_h^3 - f_\ell^3)$$

$$A_1 = \frac{1}{2}(f_h^2 - f_\ell^2)$$

where A_1 is the area of Ω_1. Similar results can be derived for Ω_2 and Ω_3.

The quantity that is analogous to a center frequency u_0 is given by

$$u_0 = \sqrt{\overline{f}_0^2 + \overline{f}_1^2} \tag{66}$$

which might be a suitable value for Equation (62). However we found experimentally that this value generated rather small windows that did not introduce enough smoothing in the low frequency channels.

In order to generate larger windows, we used the following approach to derive σ_s. Since we have a separate centroid frequency for each dimension, we first form [as in Equation (62)],

$$\sigma_{s,0} = \frac{1}{2\sqrt{2}\ \overline{f}_0}, \tag{67}$$

and

$$\sigma_{s,1} = \frac{1}{2\sqrt{2}\ \overline{f}_1}. \tag{68}$$

From here we define the parameter σ_s as

$$\sigma_s = \sqrt{\sigma_{s,0}^2 + \sigma_{s,1}^2}. \tag{69}$$

For our purposes, we found that this definition of σ_s works very well as will be shown later.

4.2.4 Texture Segmentation using the Bamberger DFB

In this section we evaluate the segmentation system using the Bamberger DFB. We use the low-high decomposition from Figure 43 which was also used in our previous work [111].

Preliminary tests using the data set were done to determine the behavior and trends of the system. However, we found quickly that the segmentation performance was poor with respect to the results reported in [106]. Close examination of the segmentation maps gives some insight to a possible reason for these results.

In Figure 59, we show the segmentation map for mixture (a) in Figure 56 using a two-stage ladder system with $N = 8$, $L = 18$, Parks-McClellan design, and $\omega_c = \frac{\pi}{4}$. The segmentation error is 30.84%. The important observation is that most of the classification errors are between the top and bottom textures in the mixture. These textures, raffia (top) and straw mat (bottom), are structurally very similar. On the other hand they are easy to differentiate with the bare eye because the structures of straw mat look significantly sharper than those of raffia. This implies that our eyes are also discriminating information based on scale. Hence, we should combine an multi-resolution analysis along with our directional analysis.

For this purpose, we have developed a set of Directional Pyramids (DPs) in section 2.5.2. In subsequent subsections we give a detailed report on segmentation results with these structures.

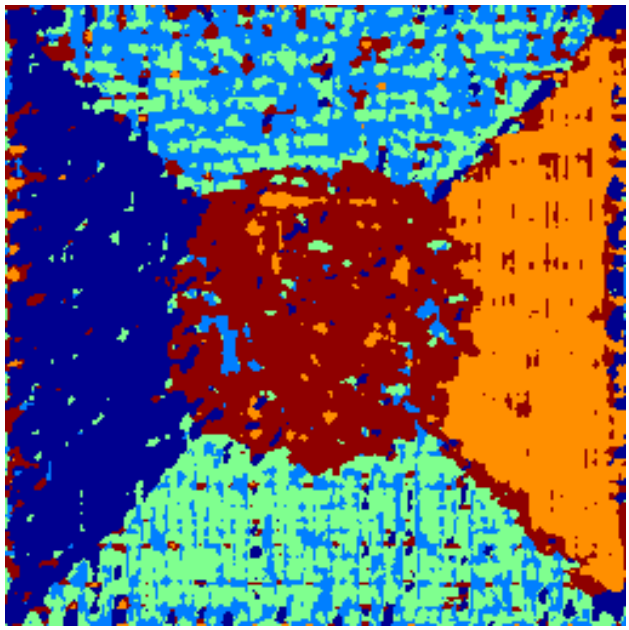


Figure 59: Segmentation map using an eight-band DFB with $\omega_c = \pi/4$

4.2.5 Evaluation of the Undecimated Directional Pyramid for Texture Segmentation

We evaluate the segmentation algorithm using the DP from Figure 46-(b). This is the ULap-UDFB pyramid where the pyramidal and directional components are undecimated. Hence full-rate subbands are available for all channels. For our experiments, we determined that four pyramid levels ($J = 4$) provided the best tradeoff-complexity tradeoff. We should note that the lowpass band was not used for feature extraction.

The evaluation is done over the remaining degrees of freedom. Namely the number of ladder stages, the ladder filter length L and design method. These parameters deal with features of the ladder structure itself. The objective in this evaluation is to examine the impact of the directional passband quality on segmentation. As we recall, for classification purposes, there is no significant improvement for large values of L.

Before continuing, we digress to evaluate the size of the codebook for the LVQ algorithm. As specified, the codebook size is 800. This codebook size increases the LVQ training time considerably, in particular during training initialization. Even on a 1 GHz pentium machine with 256 MBytes of memory, the C-code executable can take a couple of hours to train the

Table 21: Classification error as a function of codebook size

Mixture (a)		Mixture (f)		Mixture (h)		Mixture(j)		Mixture(l)	
Size	Error	Size	Error	Size	Error	Size	Error	Size	Error
5	8.52	16	94.1	10	44.58	2	1.97	2	9.81
10	5.66	80	37.48	50	23.28	10	2.04	20	4.78
100	5.32	160	30.95	100	23.85	100	1.63	100	4.56
200	5.49	240	31.31	200	22.27	200	1.57	200	4.34
400	5.67	400	29.21	400	23.22	400	1.52	400	4.60
800	5.95	800	28.66	800	22.26	800	1.70	800	4.65

system for mixtures with 10 and 16 textures.

We assess experimentally the effect of LVQ codebook size. We tested by varying the codebook size starting with the number of textures for the mixture. In the experiments we used mixtures (a), (f), (h), (j), and (l). We present the classification errors in Table 21. The DP consists of undecimated radial and undecimated angular stages. The UDFB is implemented with a two-step ladder structure with $L = 12$ and the Parks-McClellan design algorithm. We can conclude that classification error reaches a plateau for codebook sizes in the $100 - 200$ range for all tested mixtures. We decided to set the codebook size to 160 for all subsequent experiments which is the range of optimal performance for all textures while having acceptable classifier complexity.

We present the classification error results for different design method/number of ladder structures in tables 22, 23, 24 and 25. We also show the classification maps and the error maps for some of the test mixtures in Figure 60.

At the rightmost column of each table we compute the average classification error for each system. At the bottom row, we give the means of the error over a single texture. Based on these averages we give some conclusions:

1. Comparing the two-stage and three-stage ladder structures, we found that their performance is very similar. We choose the three-state ladder structures for subsequent work as they provide better passband quality.

2. We note that eight-band UDFB systems significantly outperform four-band UDFB systems.

Table 22: Classification Error for full-rate DP system with $J = 4$ and $N = 4$. The UDFB was realized with ladder structures using three ladder stages and Parks-McClellan design

L	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	Mean
4	7.02	32.00	20.19	26.77	15.24	54.93	61.31	30.68	66.67	2.94	3.00	7.18	27.33
12	5.55	30.09	19.11	26.90	16.31	52.91	59.35	28.30	68.36	2.69	3.09	6.83	26.62
18	5.33	31.16	19.33	28.05	16.75	45.18	67.65	28.63	48.25	3.02	3.08	6.82	25.27

Table 23: Classification Error for full-rate DP system with $J = 4$ and $N = 8$. The UDFB was realized with ladder structures using two ladder stages and Parks-McClellan design

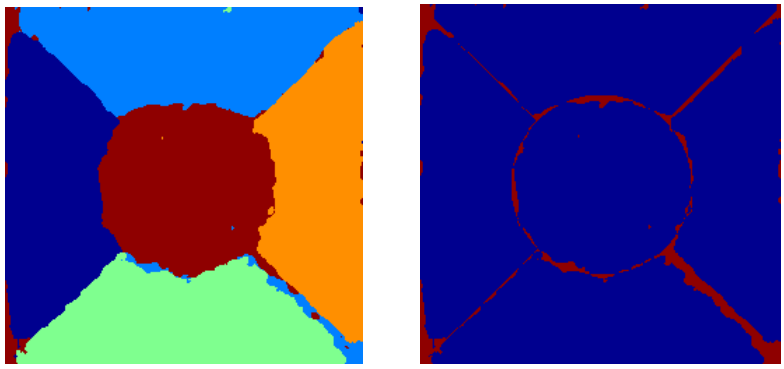
L	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	Mean
4	5.46	24.96	18.23	18.45	14.19	35.12	48.02	26.86	30.13	0.90	1.95	4.28	19.04
12	5.35	22.03	16.87	18.47	13.68	32.84	45.49	22.57	49.01	1.34	2.08	4.21	19.49
18	5.35	24.19	16.09	18.44	13.16	31.03	45.26	24.01	50.86	1.76	1.54	4.21	19.66

Table 24: Classification Error for full-rate DP system with $J = 4$ and $N = 8$. The UDFB was realized with ladder structures using three ladder stages and maximally flat design

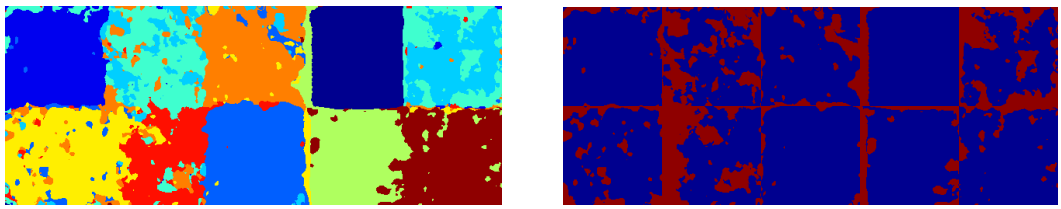
L	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	Mean
4	6.13	20.40	15.12	19.97	12.66	41.35	47.60	26.54	54.33	0.86	2.52	4.82	21.02
12	4.74	18.50	12.84	20.36	12.48	35.38	44.68	22.51	44.18	0.67	1.50	4.68	18.55
18	4.66	19.33	12.97	16.66	12.20	33.53	41.95	22.28	29.49	0.64	1.39	4.40	16.63

Table 25: Classification Error for full-rate DP system with $J = 4$ and $N = 8$. The UDFB was realized with ladder structures using three ladder stages and Parks-McClellan design

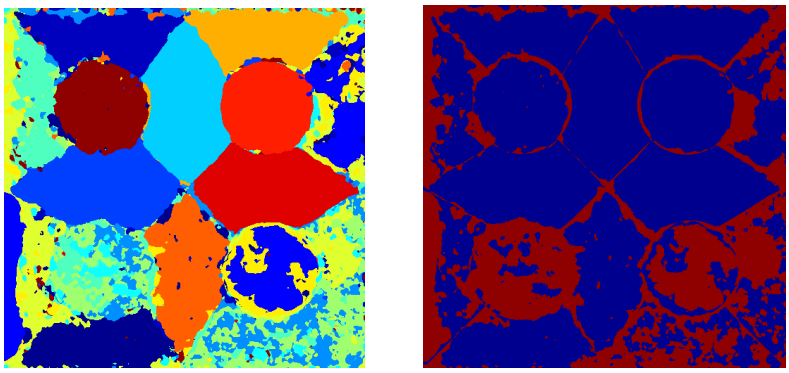
L	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	Mean
4	5.43	18.27	12.28	19.82	12.99	32.41	41.22	22.87	42.98	0.75	1.87	4.52	17.95
12	4.67	19.48	12.37	17.01	14.18	31.12	48.02	20.60	37.88	0.58	1.57	4.82	17.69
18	4.64	20.04	12.34	17.70	13.45	30.72	44.4672	20.91	29.10	0.60	1.36	4.93	16.69



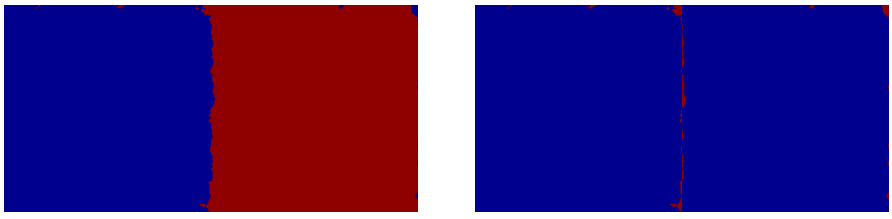
Segmentation map and error map for mixture (a)



Segmentation map and error map for mixture (f)



Segmentation map and error map for mixture (h)



Segmentation map and error map for mixture (j)

Figure 60: MR-DFB segmentation maps for results of table 25 with $L = 12$

3. The systems based on Parks-McClellan design give somewhat better performance than the maximally-flat systems. The average of the segmentation errors for each value of L shows that Parks-McClellan systems have more consistent behavior as L is varied, while maximally-flat designs show more sensitivity to this parameter. Moreover, in some cases large L works marginally better that smaller L.

4. The overall best system is from table 24 with $L = 18$. The mean classification error is 16.63%. We should note that this is a system using maximally flat filters. However as stated before, Parks-McClellan filters give more consistent performance as a function of L.

5. We should emphasize that most systems have nearly the same performance, and that this measure does not reflect the fact that some systems performed better on a texture by texture case. In fact, the best system using Parks-McClellan design has a mean classification error of 16.69% with $L = 18$ which is virtually the same as the best maximally-flat system.

Because of the performance consistency as L is varied, we would tend to recommend ladder-based UDFBs whose step filters are designed using the Parks-McClellan algorithm.

A comparison analysis with other multichannel approaches is presented later in this chapter. First we look at the impact of working with decimated systems.

4.2.6 Evaluation of decimated Pyramidal DFB systems for texture segmentation

The undecimated DP segmentation system from the last subsection requires a 24-fold data expansion (assuming the same numerical precision between input data and subband data). Hence, any possibility to reduce the computational and storage requirements is desirable. The decision to initially use full-rate systems is based on the results in [106] where it was observed that full-rate systems work significantly better than systems using decimated channels. However, since the performance of the DP systems is very good compared to other filter banks, it might be worthwhile to check the performance of the decimated DPs. The

tradeoff between decrease in complexity vs. performance degradation might be attractive enough to make the decimated systems acceptable alternatives.

To consider a decimated DP, a few modifications to our previous system are expected. We chose the ULap-DFB decomposition from section 2.5.2. We fix $J = 4$ and each radial channel is then decomposed with a maximally decimated N-band DFB. We chose this DP as the subbands across resolutions are the same size for a given orientation. This feature is useful to establish spatial relationships at the time of extracting feature vectors, as will be shown next.

In this case, the analysis DP stage has an expansion factor of three as only three undecimated radial channels are considered. The DFB stages are maximally decimated and therefore do not increase data rate. The reduction in data with respect to the undecimated system is eight-fold. Moreover, the computational reduction of the feature extraction and classification stages is more significant because of the non-linear nature of these blocks.

To form the feature vectors, we require obtaining the local energy maps from each subband. Since downsampling generates frequency aliasing, the computation of σ_s in Equation (62) needs to be re-considered. In Equations (67) and (68) we used the subband centroid $(\overline{f}_0, \overline{f}_1)$ to get Gaussian function parameters $\sigma_{s,0}$ and $\sigma_{s,1}$. If the DFB stage has an overall diagonal downsampling matrix

$$D = \begin{bmatrix} D_0 & 0 \\ 0 & D_1 \end{bmatrix},$$

then the centroid frequencies along each dimension are modified as

$$\overline{f}_{D_0} = (D_0 \overline{f}_0) \mathrm{mod} \pi \tag{70}$$

$$\overline{f}_{D_1} = (D_1 \overline{f}_1) \mathrm{mod} \pi. \tag{71}$$

The Gaussian function parameters $\sigma_{s,0}$, $\sigma_{s,1}$ and σ_s are computed on a similar fashion as in the previous subsection. However, we should note that the idea of centroid is no longer valid in this case, since aliasing causes a rearrangement of the subband support in the $[-\pi, \pi)^2$ frequency cell.

Next the feature vectors are formed from the energy maps

$$e_k^\ell(i,j),$$

where (i,j) is the spatial location within the map, $k = 1, 2, \ldots N$ is the subband index, and $\ell = 1, 2, 3$ is the pyramid level index. For a four-band DFB, the overall analysis downsampling matrix is

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},$$

so the energy maps are equally sized. Then the feature vector at location (i,j) and level ℓ is given by

$$\mathbf{f}_{i,j}^\ell = [\ e_1^\ell(i,j) \quad e_2^\ell(i,j) \quad e_3^\ell(i,j) \quad e_4^\ell(i,j)\ ]^T. \tag{72}$$

For the eight-band DFB case, obtaining the feature vectors is not as straightforward. We recall that in this case (section 1.2) there are two overall downsampling matrices: $\begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$ and $\begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$. This implies that for an $N_r \times N_c$ size image, half of the subbands are $\frac{N_r}{2} \times \frac{N_c}{4}$, while the other half are $\frac{N_r}{4} \times \frac{N_c}{2}$.

The formation of feature vector is ambiguous in the sense of how all the subbands have to be spatially related. To deal with this issue, we use the following approach: after forming the energy maps $e_k^\ell(i,j)$, $(k = 1, 2, \ldots, 8)$, we form a new set of "averaged" energy maps given by

$$a_k^\ell(i,j) = \frac{e_k^\ell(2i,j) + e_k^\ell(2i+1,j)}{2} \tag{73}$$

for $k = 1, 2, 3, 4$, $i = 0, 1, 2, \ldots, \frac{N_r}{4}$, $j = 0, 1, 2, \ldots, \frac{N_c}{4}$ and,

$$a_k^\ell(i,j) = \frac{e_k^\ell(i,2j) + e_k^\ell(i,2j+1)}{2} \tag{74}$$

for $k = 5, 6, 7, 8$, $i = 0, 1, 2, \ldots, \frac{N_r}{4}$, $j = 0, 1, 2, \ldots, \frac{N_c}{4}$. This step allows us to obtain equal size energy maps, and the feature vector for level ℓ is given by

$$\mathbf{f}_{i,j}^\ell = [\ a_1^\ell(i,j) \quad a_2^\ell(i,j) \quad \ldots \quad a_8^\ell(i,j).\ ]^T. \tag{75}$$

With this averaging operation we potentially lose spatial information; on the other hand we also gain simplification in classifier complexity by a factor of 2. The final feature vectors

Table 26: Classification Error for directional decimated DP system with $J = 3$ and $N = 4$. The maximally decimated DFB was realized with ladder structures using three ladder stages and Parks-McClellan design

L	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	Mean
4	5.51	25.98	20.65	26.39	17.96	55.35	50.96	44.77	34.87	7.07	3.46	2.89	24.65
12	4.90	26.51	19.29	25.74	17.81	48.76	50.98	27.09	35.92	7.44	2.84	2.39	22.47
18	5.61	25.75	20.48	25.63	16.13	50.26	50.54	26.54	35.60	6.87	3.19	2.30	22.41

(for either four or eight DFBs) are obtained by concatenating the feature vectors for each pyramid level. Namely,

$$\mathbf{f}_{i,j} = \begin{bmatrix} \mathbf{f}^1_{i,j}(i,j) \\ \cdots \\ \mathbf{f}^2_{i,j}(i,j) \\ \cdots \\ \mathbf{f}^3_{i,j}(i,j) \end{bmatrix}. \tag{76}$$

Finally, the classifier produces a segmentation map $S(i,j)$ that is smaller than the original image size according to the analysis downsampling matrix. A classification label at location (i,j) corresponds to an $N \times N$ area of the original texture $x[n_0, n_1]$ with the same spatial orientation (N is the number of bands in the DFB). To generate a segmentation map the same size as $x[n_0, n_1]$, we first generate the Kronecker product

$$S'(n_0, n_1) = S(i,j) \otimes \mathbf{1}_{N \times N}, \tag{77}$$

where $\mathbf{1}_{N \times N}$ is an $N \times N$ matrix of ones. This operation generates jagged artifacts around the texture boundaries. To reduce the artifacts, we apply a 3×3 median filter to $S'(n_0, n_1)$. The final segmentation map is then given by

$$S''(n_0, n_1) = \text{median}(S'(n_0, n_1))_{3 \times 3}. \tag{78}$$

With these modifications in place, we proceeded to evaluate the new segmentation system similar to the full-rate case. Again, we set the number of pyramid levels to 3. We show the results in tables 26, 27, and 28.

First we note that the systems with $N = 8$ outperform the $N = 4$ systems. An important point to keep in mind is that for $N = 8$ we have half the spatial information than for $N = 4$.

Table 27: Classification Error for directional decimated DP system with $J = 3$ and $N = 8$. The maximally decimated DFB was realized with ladder structures using three ladder stages and maximally flat design

L	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	Mean
4	7.25	19.90	17.08	21.50	12.52	36.24	47.85	25.34	28.25	2.33	1.48	3.27	18.58
12	6.79	19.36	16.44	16.79	12.90	31.62	45.41	19.62	25.02	1.22	1.36	2.50	16.59
18	6.31	17.69	16.04	15.42	12.66	30.30	45.70	19.23	26.26	1.29	1.41	2.37	16.22

Table 28: Classification Error for directional decimated DP system with $J = 3$ and $N = 8$. The maximally decimated DFB was realized with ladder structures using three ladder stages and Parks-McClellan design

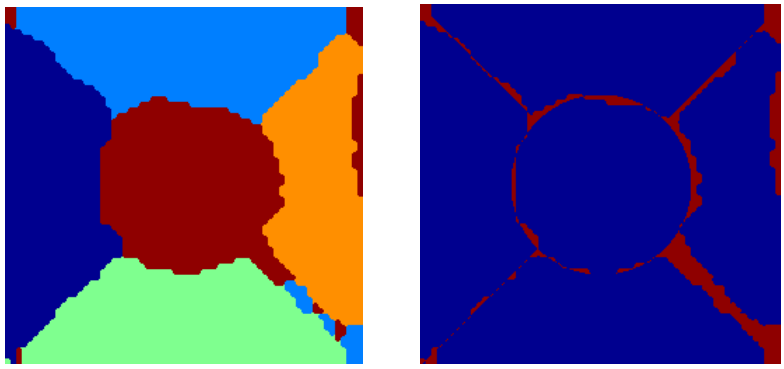
L	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	Mean
4	5.57	19.20	15.08	19.27	12.66	32.79	44.01	20.35	26.05	1.08	1.52	2.61	16.68
12	6.46	17.72	14.95	16.43	12.05	29.36	44.62	18.02	26.77	1.09	1.35	2.37	15.93
18	6.78	18.45	15.51	15.09	12.33	30.19	43.84	17.99	26.55	1.18	1.36	2.32	15.97

Therefore, we conclude from the experimental evidence that directional selectivity is a more important property than spatial selectivity. Other decimated systems like the 2-D DWT and uniform 2-D filter banks have limited directional selectivity.

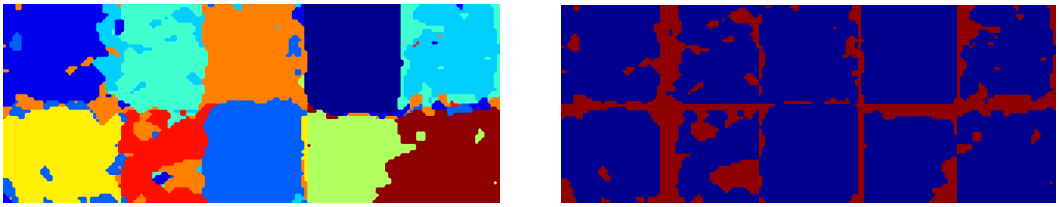
The second crucial result is that we see that decimated systems work as well as undecimated systems. In fact, the decimated DFB-based systems perform slightly better on the average as can be concluded from the tables. The best performance is given by the system in Table 28 with $L = 12$ with an average segmentation error of 15.93. Although the decimated pyramidal DFB is not maximally decimated, it works with substantially less data than the full-rate case, and is closer in complexity to maximally decimated systems.

Visually, we present the segmentation results for the decimated system in Figure 61. We note that the maps are very close to those of Figure 60. The main difference is on the delineation of boundaries which is (as expected) less well defined in the decimated case. On the other hand, in some cases the internal regions are better identified by the decimated system.

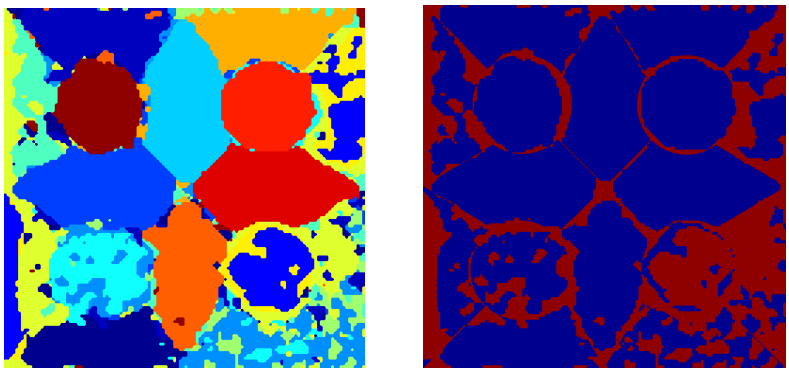
We speculate that the good performance of the decimated system is caused by a couple of mechanisms working in conjunction. First, as noted above, the good directional selectivity provides good discrimination among the features. Second, the downsampling operations could reduce the local texture variations which produces more uniform regions on the local



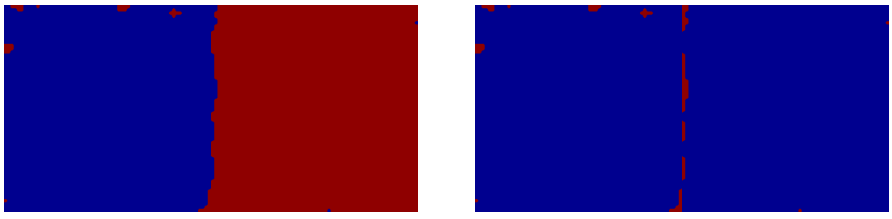
Segmentation map and error map for mixture (a)



Segmentation map and error map for mixture (f)



Segmentation map and error map for mixture (h)



Segmentation map and error map for mixture (j)

Figure 61: MR-DFB segmentation maps for results of table 28 with $L = 12$

energy maps. This last observation is consistent with the blob detection hypothesis developed by Jain and Farrokhnia in [56], and with the classness vs. resolution argument cited by Wilson and Spann in [135].

In other decimated multichannel systems reported in [106], the energy maps of the lower rate subbands were upsampled to full rate, in order to deal with non-uniform sized channels, The upsampling process could be as simple as sample replication or as complex as using an anti-imaging filter. It seems that upsampling the subbands hinders the discriminative abilities of the feature set, probably as a result of aliasing introduced by upsampling.

4.2.7 Comparison with previous work

Finally, in this section we compare the segmentation results based on our DPs with other multichannel schemes. An extensive number of filter banks were evaluated in [106]. The mean classification error over the texture mixtures is presented in table 29. (Note that these results reflect a correction on the results of texture mixture (i) as reported in [105].) The following general conclusions were reported:

- Full-rate systems outperform decimated systems.
- No approach is clearly superior to the others.
- For wavelet and filter bank decompositions, wavelet packets provide better performance than dyadic representations.
- The "best" system is the 16-tap Johnston QMF filter bank with 40 subbands. The system is full-rate, so complexity is a significant drawback.
- For two-texture mixtures, optimized methods are better, however for more complex images performance degrades quickly.
- The best tradeoff between complexity and performance is given by the DCT-based system which achieves 24.85% average classification with only 8 subbands.

The results for the DT-CWT from de Rivaz [109] are also reported in table 29. This decomposition gives a clear improvement from the rest of the filter banks. The DT-CWT

Table 29: Average classification errors for different segmentation systems

Segmentation Method	Mean Classif. Error	Segmentation Method	Mean Classif. Error
Laws Filters	28.32	Co-occurrence	29.1
Ring/wedge filters	30.6	Autoregressive	26.32
Dyadic Gabor	27.77	Eigenfilter	26.17
Gabor filt. bank	31.45	Prediction error	29.27
DCT	24.85	Optimized FIR J_u	26.36
Daub. 4 (UDWT)	22.55	Back Prop. NN	51.92
Daub. 4 (DWT)	28.55	DT-CWT [109]	18.0
QMF 16-tap	20.83		

Table 30: Comparison of segmentation results for different multichannel approaches

Description	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)	Mean
DCT	13.2	27	25.5	37.8	22.6	40.9	49	38.2	33	6.4	2.2	2.5	24.85
FR-QMF	8.7	18.9	23.3	18.4	17.2	36.4	41.7	39.8	28.5	8.1	0.8	8.2	20.83
SPyr	7.1	26.2	24.3	29.9	20.2	42.27	63.92	35.29	43.31	2.99	1.91	12.98	25.87
DT-CWT	10.9	21.8	16.2	16.6	17.3	33.8	40.4	19.3	28.6	0.6	1.1	9.3	17.99
ULap-UDFB, L=18	4.66	19.33	12.97	16.66	12.20	33.53	41.95	22.28	29.49	0.64	1.39	4.40	16.63
ULap-DFB L=12	6.46	17.72	14.95	16.43	12.05	29.36	44.62	18.02	26.77	1.09	1.35	2.37	15.93

is nearly shift-invariant, and has twice the directional selectivity of the classical DWT. De Rivaz [109] claims that the reason for this improvement is caused by having complex-valued channels which reduce the oscillatory fluctuations from the rectifying non-linearity output.

For further comparisons, we use the two multichannel decompositions recommended by Husøy and Randen in their work: the DCT and the full-rate non-uniform QMF filter bank with 40 subbands. We also use the DT-CWT results. We present the classification errors for all the texture mixtures in table 30.

Additionally, we implemented a segmentation system using the Steerable Pyramid (SPyr) from Simoncelli, et al. [118]; to the best of our knowledge these results have not been reported. We use four pyramid levels combined with an eight-band angular component. The lower rate energy maps where upsampled using a combination of zero-order interpolation followed by a 3×3 median filter. The highpass residual was included for feature extraction.

On our side, we have selected the systems that provide the lowest average classification error. We should note, however, that all tested ULap-DFB and ULap-UDFB systems give similar segmentation performance.

Using the average of the classification errors as a single metric for performance, we see that the systems proposed in this thesis are the top performers. Using the average classification error oversimplifies the picture. It might be more useful to look at the mixtures on a case by case basis. Under this metric, we see that except for mixtures (g), (j), and (k), that our systems provide the best results. Moreover, the DT-CWT performs marginally better on mixtures (g),(j), and (k). It should be noted that the differences in classification error are marginal among the top performers, and that their segmentation maps will not have visually discernable differences.

The steerable pyramid is the worst performer. Similar to other systems with decimated channels, it seems that upsampling the lower rate energy maps to full rate does not improve discriminative power. Working at the lower rates as described in the previous section accounts better for the energy distribution of textures despite the loss in spatial resolution.

The Lap-DFB and ULap-UDFB pyramids and the DT-CWT representations are very close in performance. However the DFB-based systems usually outperform the DT-CWT. To further differentiate among the two approaches it is necessary to look into other aspects of the representations. As far as complexity, the DT-CWT (4-levels, 6-orientations) is comparable to the Lap-DFB system (3-levels, 8-directions). The DT-CWT is redundant by a factor of four, while the decimated DP increases the data by factor of three. Hence, there is advantage on using the Lap-DFB. Also, the computation of features from DT-CWT is slightly more involved, since the local energy calculation has to be computed from complex-valued subbands.

In [109] it is concluded that the improved directionality of the DT-CWT is the principal mechanism behind the improvement in segmentation quality, and that the complex-valued nature of this representation is not relevant. In fact, they show that the real part of the DT-CWT performed as well as the full DT-CWT. This result supports our own conclusions about the DFB-based systems: increased directional resolution improves the discriminative power of the local-energy features despite the loss in spatial resolution caused by downsampling.

Finally, for segmentation purposes shift-invariance does not seem to be a relevant requirement. The directional bands of the ULap-DFB are maximally decimated, and nonetheless it achieves the highest performance.

On the full-rate front, we found the ULap-UDFB pyramid (with 24 subbands) significantly outperformed the 40 subband undecimated QMF system. The main difference between the two systems is again the improved directional selectivity compared to the traditional 2-D DWT limited to vertical, horizontal and diagonal detail. However, since the ULap-DFB pyramid systems work as well as their full-rate counterparts, we recommend using the former because of the obvious advantage on computational and storage requirements.

Given that directional resolution is the significant factor on improving segmentation, the ULap-DFB is the best suited representation for this task. This structure is not limited to a fixed number of directions, and we can easily increase the number of directional subbands by factors of two (or by other factors as reported in [5]). Furthermore, this family of DPs offers a lot of flexibility regarding the filter passband characteristics and design method. Finally, the efficient polyphase domain DFB structure makes the DFB-based systems very attractive for real world (hardware and software) implementations.

More specifically, we found that an eight-band DFB implemented with three ladder stages gave the best performance on the test set compared to all other systems. Using the Parks-McClellan filter design method improves segmentation over the maximally-flat design method on the average. The length of ladder filters $\beta(z)$ had no definite influence on performance, hence in general we can expect good performance with very small filters.

4.2.8 Conclusions on Texture Segmentation

In this section we have done an extensive evaluation of UDFB and DFB-based pyramid structures as tools for texture segmentation. We can report that the need of a pyramidal component is critical to achieve good performance. We found that the ULap-DFB structure provides the best results reported to date on the test set we have used. Excellent

performance can be obtained using decimated DFBs as the angular component of the directional pyramid. This has not been the case for other decimated image decompositions. The additional orientation selectivity of our system seems to be the principal factor for such improvement. Also, the degrees of freedom provided in our systems, allows us to search for the set of parameters that would provide the best performance for a specific application.

4.3 Texture Modeling Using DFB-Based Pyramids

A Texture analysis/synthesis algorithm introduced by Heeger and Bergen [49] has been the basis of extensive work on texture modeling. They combine the use of first order spatial statistics (i.e. histograms or distributions) with the use of directional multiresolution analysis. The algorithm is based on findings about human perception, but is not a complete model since it only relies on first order statistics.

The algorithm is presented in Figure 62. The first step is to calculate a directional pyramid decomposition of the texture sample. Second, the histograms of the texture sample and the subbands are obtained. Third, we generate a Gaussian noise 2-D field and match its histogram to that of the texture sample. Next, the multichannel decomposition of the noise is calculated, and the histograms of the subbands are matched to those of the texture sample. The pyramid is collapsed, and the full procedure is repeated a few times. The resulting texture is visually similar to the original. Such a model is very limited in regards to the types of textures it can represent. The Heeger-Bergen algorithm produces good results on some type of stochastic (i.e. random looking) textures, however it fails to model textures with some degree of regular structure.

No proof of convergence for the algorithm has been given. However, after about 5 iterations, a good visual approximation is obtained. Second, an multi-resolution analysis is done using the Steerable Pyramid from Simoncelli, et al. [118]. The steerable pyramid does a better job of capturing textural features due to its ability to decompose an image in directional subbands across scales, similar to the way our visual system represents information.

In this section, we focus on evaluating the use of DFB-based pyramids as an alternative to the steerable pyramid. We look at the maximally decimated DFB in order to achieve

```
1: function MatchTexture(texture,synthTexture)
2:     Input: texture                                  ▷ texture to be modelled
3:     Output: synthTexture                            ▷ synthesized texture
4:
5:     analysisPyr = MakePyramid(texture);
6:     synthTexture = RandomNoiseGen(); ▷ Generate a 2-D random noise field
7:     synthTexture = MatchHistogram(synthTexture,texture);
8:
9:     for s doeveral iterations
10:        synthesisPyr = MakePyramid(synthTexture);
11:        for i do = 1:numberOfSubbands
12:            Sband_i = MatchHistogram(Sband_i,Aband_i);
13:        end for
14:        synthTexture = CollapsePyramid(SynthesisPyr);
15:        synthTexture = MatchHistogram(synthTexture,texture);
16:    end for
17: end function
```

Figure 62: Pseudo-code for the Heeger-Bergen texture synthesis algorithm

computational and storage savings with respect to the steerable pyramid.

Using UDFB-based structures could require less computation than the steerable pyramid, but the data expansion would be of similar order. Nonetheless, UDFB-based pyramids are also valid decompositions for the Heeger-Bergen algorithm. In section 6.4, we use an overcomplete UDFB pyramid for the removal of speckle noise in SAR images.

Our first approach was to use the DFB structure on Figure 43. Here we also need to choose the cutoff frequency ω_c of the lowpass filter. We quickly found that the quality of the synthesized textures was very poor. This should be expected since as described in [49], the subband histograms are an incomplete model. Moreover purely directional representation would be incapable of capturing of distinguishing texture components at different scales.

Hence, it is necessary to get a better separation of texture elements from the pyramid decomposition. We use one of the directional pyramids described in section 2.5.2. We study two structures the Lap-DFB pyramid and the ULap-DFB structure. As explained, the difference is that in the former structure, the lowpass image is a downsampled at each level of the pyramid. On the latter case, the bandpass components are undecimated, hence each bandpass is the same size as the original, and the directional subbands have the same dimensions for all pyramid levels.

Some experimentation showed that the ULap-DFB captured much better the detail of the test images. Moreover, we found experimentally that the decomposition with four levels and eight directional bands gives the best visual synthesis. The Lap-DFB structure gives very poor visual results. This seems to be related to the smaller amounts of data on the lower resolution subbands, which leads to histograms incapable of modelling textural features.

For the ULap-DFB we used four resolution levels and an eight-band DFB on each detail level. The DFB was implemented with a two-stage ladder structure using step filters of length 18 designed with the Parks-McClellan method. For the steerable pyramid we used four resolution levels and four directional subbands for each detail level.

We show in Figure 63 some successes on synthesis using both the steerable pyramid and the DFB. We observe good visual similarity among the two approaches with respect to the original sample. In Figure 64, we see failures on textures that have fine structures. In these cases, it is interesting to note that some of the basic elements of the texture are obtained. However, the algorithm cannot capture the positioning rules for the primitive elements.

The objective of this section was to show an alternative to the image pyramid used in the Heeger-Bergen algorithm. We showed experimentally, that the visual quality of synthesis is as good as that of the steerable pyramid with much less information required. It seems that the increased directional selectivity offered by the DFB is a determinant on capturing the textural information at a fraction of the resources. Future work should be directed to experiment with more complete texture models based on pyramid decompositions, like those described in Zhu, et al. in [142], and Portilla and Simoncelli in [103].

4.4 Chapter Summary

The main contribution of this chapter is the use of directional pyramids for applications on texture analysis and synthesis. Although the ability to discern directional information is important, we found that we cannot ignore the scale component. Specifically, we used the ULap-DFB and ULap-UDFB pyramids described in Chapter 2.

In particular, we studied rotation invariant texture classification, texture segmentation,

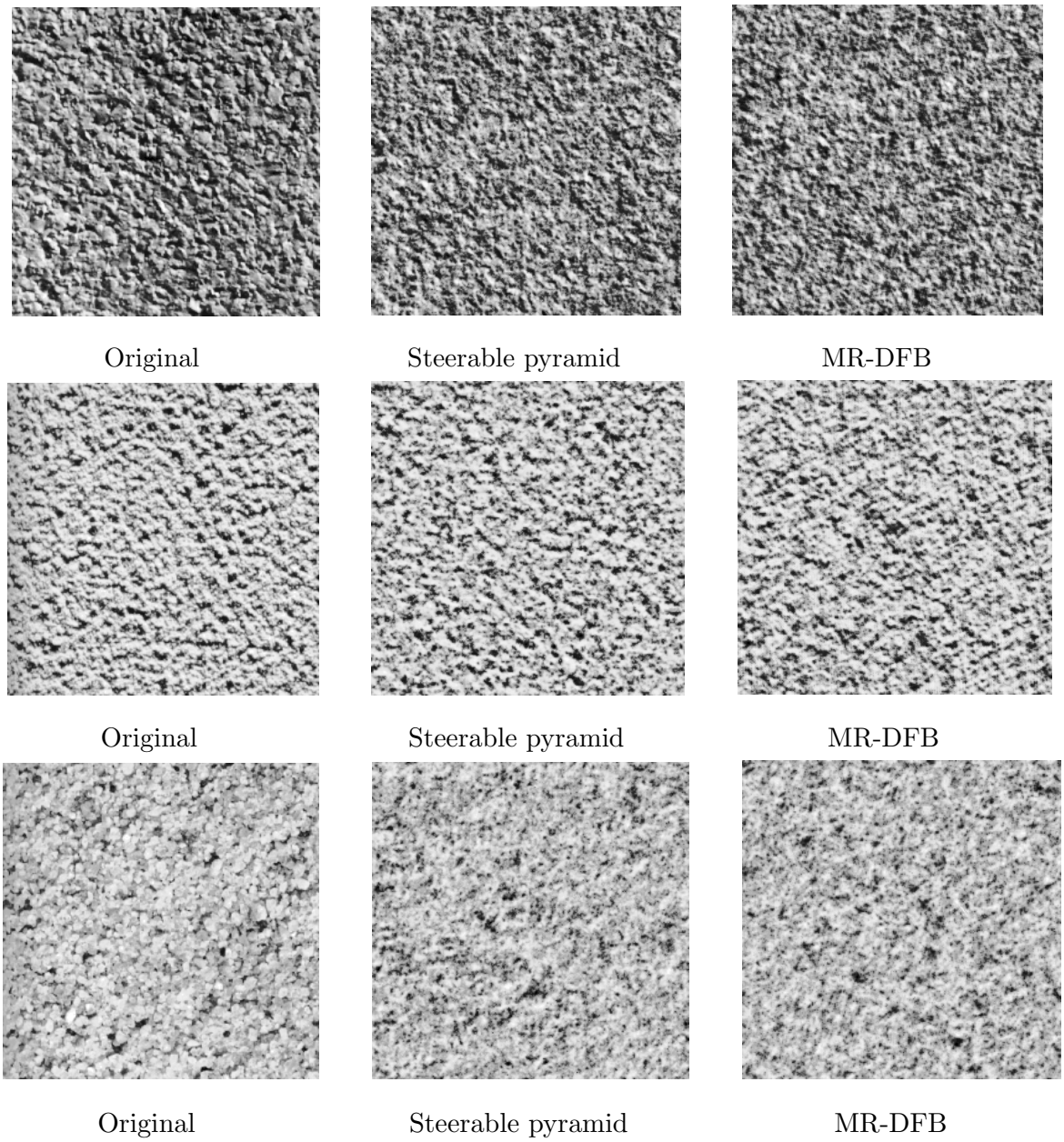


Figure 63: Examples of successful texture synthesis with the Heeger-Bergen algorithm

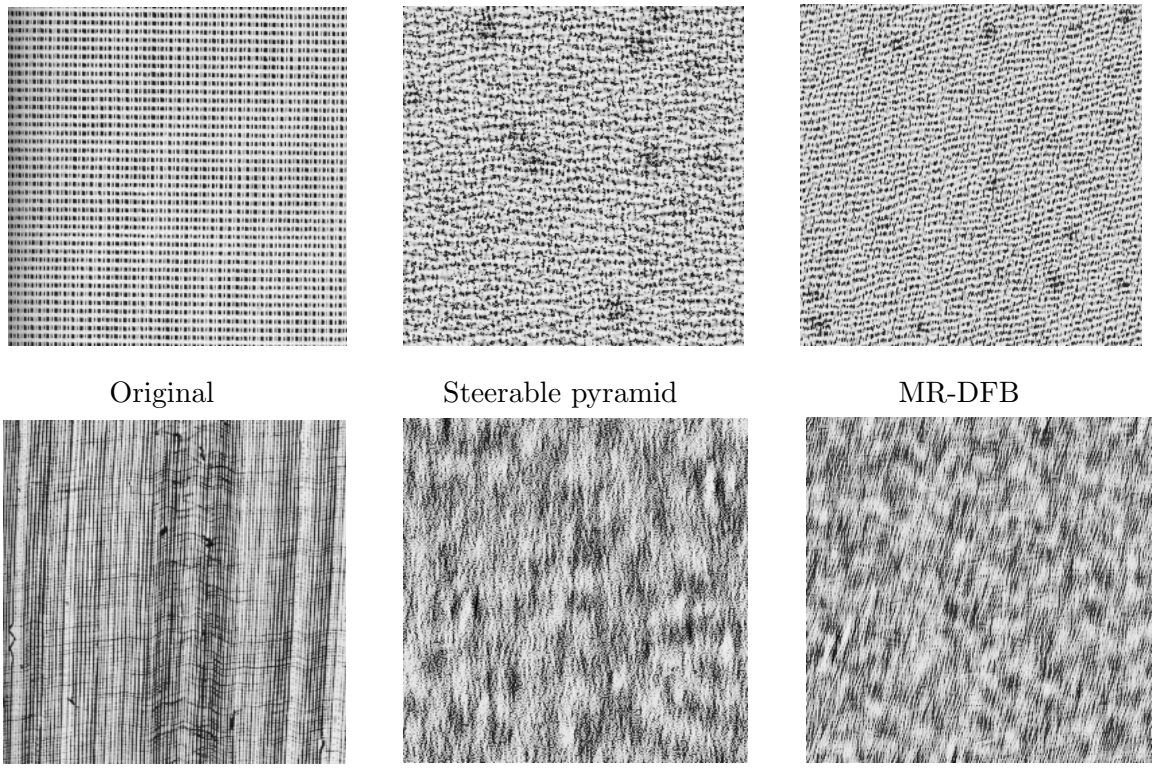


Figure 64: Examples of texture synthesis failure with the Heeger-Bergen algorithm

and texture synthesis. On these three problems we presented results that are either superior or very competitive with respect to previously reported work. The experimental evidence shows that these gains are related to the quality on the angular information provided by the DFB and UDFB structures. Further work should focus on developing DFB-based algorithms for specific applications.

CHAPTER V

IMAGE DENOISING

In this chapter we adapt the wavelet denoising schemes to the DFB-based pyramids presented in Chapter 2. As is well known, the 2-D DWT has a limited orientation selectivity which provides poor localization of edges that are not aligned along the vertical, horizontal or diagonal axis.

A natural step is then to use representations that capture angular information with more precision in order to provide better isolation against noise. The premise would be that by capturing directional information more accurately, we can obtain higher edge sharpness than the DWT after the denoising process. We will show in the following sections that this is the case.

5.1 Denoising via wavelet thresholding

An active area of research has been the removal of noise in signals using wavelet domain thresholding. The method is simple and provides superior performance to the optimal linear filtering methods. The scheme has been used for 1-D and 2-D signals.

Given a discrete signal $y[n] = x[n] + \varepsilon[n]$ where $x[n]$ is the signal we want to estimate and ε is additive white noise with $n \sim \mathcal{N}(\prime, \sigma)$, an estimate of $x[n]$ can be obtained with the following method

1. Calculate the 2-D DWT of y. $Y = \mathcal{W}y$.
2. Classify each wavelet coefficient as noise or signal (i.e., apply a threshold $\hat{Y}_i = \eta(Y_i, t_i))$).
3. The estimate of the clean signal x is given by $\hat{x} = \mathcal{W}^{-1}\hat{Y}$.

This approach is closely related to wavelet compression (i.e. subband coding) schemes, and relies on the localization property of the wavelet transform. Since $Y = X + E$, where

$X = \mathcal{W}x$ and $E = \mathcal{W}\varepsilon$, we can look at the effects of the transform on each component.

For $x[n]$ the signal energy is concentrated on a few coefficients of large magnitude which are usually grouped on small sparse clusters. Hence most of the coefficients can be discarded without loss of the essential features of the image. This is the principle under which wavelet compression works.

In the case of $\varepsilon[n]$, since $\mathcal{W}$ is an orthogonal transform, we have that $N \sim \mathcal{N}(\prime, \sigma)$, and the noise remains white.

From the additive model, it is intuitive to see that noisy signal coefficients whose magnitude remains above σ can be considered as being more probable to belong to the $x[n]$, while all coefficients below σ could be considered noise. This simple classification scheme uses σ as a threshold, hence the term wavelet thresholding has been popularized. These empirical observations were formalized by the extensive work of Donoho and his collaborators [32, 33]. The generalization of this method to 2-D images has been studied extensively by now.

5.2 Thresholding rule

The selection of threshold or some other classification scheme has been the main focus of attention in the literature. Thresholding is comprised of a threshold rule and a threshold value, and both are usually treated as dependant of each other.

Threshold rules fall into two categories: hard and soft thresholding. For hard thresholding the rule given by

$$\hat{x} = \eta_H(x, T) = \begin{cases} 0 & |x| < T \\ x & |x| \geq T. \end{cases} \tag{79}$$

The soft thresholding rule is given by

$$\hat{x} = \eta_S(x, T) = \begin{cases} 0 & |x| < T \\ sign(x)(|x| - T) & |x| \geq T. \end{cases} \tag{80}$$

Although hard-thresholding provides a better approximation in the mean square error (MSE) sense, soft-thresholding has been preferred for a few reasons. First, Equation (80) is mathematically more tractable, as hard thresholding introduces a discontinuity. Donoho has shown that a soft threshold is optimal under a minimax criterion and that it is adaptable to

unknown smoothness [33]. For application to images soft-thresholding provides a smoother reconstruction which is more pleasant to the eye, while hard-thresholding leaves blips on smooth areas as a manifestation of the discontinuity present in this rule. Our own experience also identifies soft-thresholding as the better scheme for natural images. In the remaining discussion of this chapter we will use soft-thresholding.

5.3 Image denoising

As natural images are non-stationary signals, estimating an overall threshold makes no sense. One approach commonly taken is to estimate a separate threshold for each subband. A good summary of this approach is presented by Chang, et al. in [17]. They evaluate different subband threshold estimation algorithms and propose the *BayesShrink* threshold. This threshold is found by minimizing

$$T^* = \arg\min_T E_{Y|X,X}(\eta_T(Y) - X)^2 \tag{81}$$

assuming that the clean image coefficients X have a Generalized Gaussian (GG) distribution and $Y|X \sim \mathcal{N}(\dagger - \S, \sigma)$. Rather than performing this minimization, it was found that the following approximation works for a wide range of generalized GG distributions

$$T_B = \frac{\sigma^2}{\sigma_x} \tag{82}$$

where σ is can be estimated by

$$\hat{\sigma} = \frac{\text{MEDIAN}(|y[i,j]|)}{0.6745} \tag{83}$$

where $y[i,j]$ refers in the case to the HH subband at the finest scale. Similarly, σ_X can be estimated by

$$\hat{\sigma_x} = \max(\sigma_y - \sigma, 0). \tag{84}$$

Obviously the variance of y can be estimated from the subband data. This threshold showed top performance while being extremely easy to calculate.

The 2-D nature of images allows us to explore thresholding methods that exploit 2-D relationships. A few algorithms have been proposed in this front. Jansen and Roose [58] has

proposed a Bayesian approach using local geometrical models and a Gibbs sampler. Mihcak, et al. [87] have done spatially adaptive statistical modelling of the subband coefficients. Several other locally adaptive schemes have been proposed. Next we focus our attention on the scheme by Chang, et al. [18] which is considered a comparison benchmark as it provides some of the best results to this date.

5.3.1 The SAWT algorithm

The Spatially Adaptive Wavelet Thresholding (SAWT) algorithm from [18] has provided some of the best results on wavelet image denoising to this date. It is based on modelling the local interactions of subband coefficient $y_s[i_o, j_0]$ with its eight nearest neighbors coefficients. In addition to the local interactions, the child-parent subband interactions are modelled by the relation of $y_s[i_o, j_0]$ with $y_{s+1}[\lfloor i_o/2 \rfloor, \lfloor j_0/2 \rfloor]$, where s represents the subband resolution.

These interactions are modelled by calculating a context value for each location $[i_0, j_0]$. The context $Z[i_0, j_0]$ is calculated by taking the simple average of the eight neighbors and the parent coefficient.

The threshold estimate is a modification of the $BayesShrink$ threshold of Equation (82) and is given by

$$T_B[i_0, j_0] = \frac{\sigma^2}{\sigma_x[i_0, j_0]} \tag{85}$$

To estimate of $\sigma_x[i_0, j_0]$ is now done with the K values of $y[i, j]$ whose context values $Z[i, j]$ are closest to $Z[i_0, j_0]$. In other words, $\sigma_x[i_0, j_0]$ is calculated with data that is statistically more similar. This procedure is repeated for all, but the coarsest, subbands in the image.

Finally, one more twist was added to the SAWT scheme. As is well known, maximally decimated transforms lack shift invariance. This introduces ambiguity in the localization of edges in the subband with respect to localization of edges in the original image. Hence, SAWT was also applied to the Undecimated DWT (UDWT) in [18]. However, shift-invariance is obtained at the cost of making the noise in the subband correlated. To deal with colored noise, each subband was split into its orthogonal shifts and processed separately with SAWT. The results of the UDWT-SAWT combination are some of the best reported to date, and are commonly used to benchmark other algorithms.

In the following sections we modify SAWT so that it can be used with the DFB-based pyramids developed in this thesis.

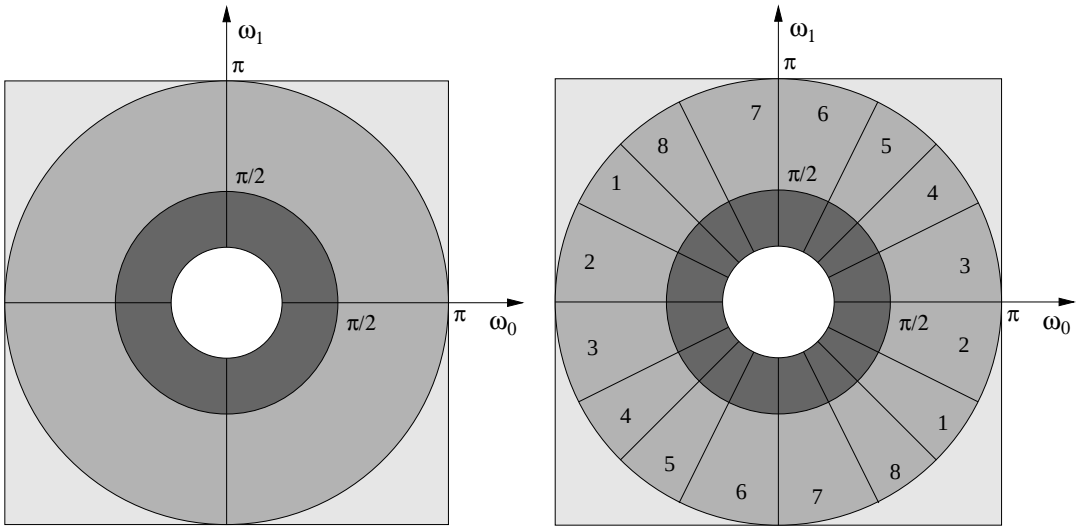
5.4 Image denoising based on the Bamberger DFB

Edges are perceptually important features which our visual system uses to assess image quality. Our eyes are extremely sensitive to the blurring of edge and fine textures. The 2-D DWT and UDWT offer poor directional resolution (only three orientations), hence it is a logical idea to explore the use of transforms which provide better directional selectivity and by implication, a better representation of edges.

In our previous work [110], we introduced the use of the Bamberger DFB for image denoising. In that work we separated the image using the lowpass-highpass decomposition from Figure 43, and then split the highpass component with an eight-band DFB. We determined empirically a soft-threshold estimate of $T_i = 1.5\sigma_i$ for each subband. With respect to the DWT, the results from the DFB system showed sharper, well preserved edge detail. However the presence of cross-hatching artifacts on smooth regions was also present.

As we saw in Chapter 4, the combination of the DFB and UDFB with a pyramidal (i.e. multiresolution) decomposition provided much better discriminatory power than using the simple lowpass-highpass decomposition. We would expect similar improvements in the denoising problem, as edge detail can be differentiated across scales with respect to the strength of the noise. Moreover, the advantage of retaining shift-invariance on the subbands is now well understood(e.g., the UDWT) [18].

Hence, in this chapter we adopt a directional pyramid based on the UDFB in order to retain shift invariance across orientations. We saw in our segmentation work that an undecimated radial pyramid provided unambiguous feature location over the image plane, so that all features across resolutions could be properly aligned. On the other hand, the use of an undecimated radial pyramid increases the amount of data significantly. For denoising, each subband can be processed separately or with the dependance on only the parent subband as with the SAWT algorithm. Hence, it is possible to use a radial decomposition with a smaller storage requirement like the Laplacian pyramid [14]. In this case, orientation



(a)Radial Component (b)Radial-Angular Partitioning

Figure 65: Frequency plane partitioning used for the OCR-UDFB and the Steerable Pyramid.

shift invariance is retained at each resolution level.

An alternative to the Laplacian pyramid is the radial decomposition used by Simoncelli, et al. [118] for the steerable pyramid (SPyr). The frequency partitioning for this pyramid is shown in Figure 65-(a). In this case, the filters have a 2-D circular support, and there is an additional band for frequencies $\sqrt{(}\omega_0^2 + \omega_1^2) > \frac{\pi}{2}$. Each resolution level is downsampled by a factor of 2 on each direction. As explained by Simoncelli to the author, this additional residual highpass band decreases aliasing introduced by the downsamplers. We call this pyramid the over-complete radial (OCR) pyramid in order to differentiate it from the undecimated version where no downsamplers are used.

For the following work, we will form a new polar-separable pyramid using the OCR Figure 65 in combination with an eight-band UDFB. Hence, the OCR-UDFB pyramid only differs from the steerable pyramid in the directional component of the decomposition. This will allow us to do also compare both decompositions. The frequency plane partitioning for the OCR-UDFB is presented in Figure 65-(b), the highpass residual and the lowpass band are not decomposed with the UDFB.

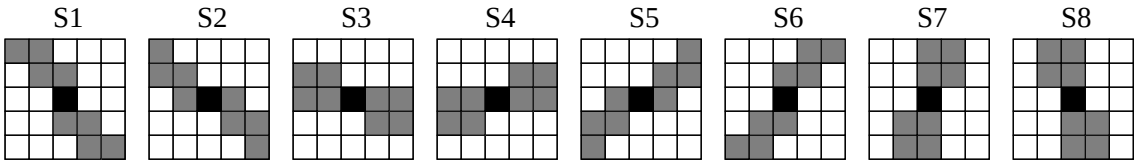


Figure 66: Definition of local neighborhoods for the SAWT algorithm. The center pixel (black) is the pixel to be thresholded. The gray pixels form the neighborhood N_ℓ for each subband orientation.

5.5 Denoising with the OCR-UDFB pyramid

First, we describe the adaptation of the SAWT algorithm to the OCR-UDFB. The threshold for each subband coefficient is determined by analyzing the structure of a surrounding local neighborhood in order to do a context-based ranking. Coefficients with similar contexts are grouped together in order to determined the *BayesShrink* threshold.

For our decomposition we found that choosing the structure of the local neighborhood so that it aligns with the orientation of the subband improves results significantly. In Figure 66 we contrast the neighborhood definitions for the UDWT and UDFB for this algorithm. Obviously, there are three different neighborhoods for the UDWT and eight for our UDFB.

With regard to the parent band, in this case it is half the size on each dimension, so each coefficient in the parent band has four children at the finer scale.

For the UDFB, we found the three-stage ladder implementation performs better than the two-stage ladder. On the other hand, the design method for the step filters and for step filters of length eight and above, there was no difference in performance. We set $L = 14$ for our experiments, and the ladder filters are designed with the Parks-McClellan algorithm. Finally, we set the number of radial levels to $J = 5$ and the number of directional bands to $N = 8$.

In Tables 31 and 32 we present simulations for the 512×512 Lenna and Barbara images for different noise standard deviations. We compare the results with the UDWT and the Steerable Pyramid (SPyr). For the SPyr we set $J = 5$ and $N = 8$, and the SAWT algorithm was modified as in the OCR-UDFB case.

We found that all systems have very close performance. For Lenna, the OCR-UDFB

Table 31: MSE results for Lenna using different image decompositions using SAWT

MSE/ σ_n	12.5	15	17.5	20	22.5	25
UDWT - Symmlet	24.9	29.9	35.5	40.2	45.2	50.8
UDWT - 9/7	24.3	29.4	34.0	39.6	44.5	49.9
SPyr	25.0	29.8	34.8	39.7	44.7	50.73
OCR-UDFB	24.4	28.8	33.6	38.6	43.6	49.4

Table 32: MSE results for Barbara using different image decompositions using SAWT

MSE/ σ_n	12.5	15	17.5	20	22.5	25
UDWT - Symmlet	39.5	50.4	60.7	73.2	85.3	96.2
UDWT - 9/7	39.8	50.7	61.6	73.6	85.7	99.5
SPyr	43.2	55.4	68.3	80.2	93.8	106.6
OCR-UDFB	40.9	51.4	62.2	74.0	85.9	98.3

system performs slightly better than the other systems, but the performance is very close to the UDWT using the 9-7 biorthogonal wavelet filters. In the case of Barbara, the UDWT systems perform slightly better than the OCR-UDFB, but the differences in MSE are insignificant. The SPyr shows higher MSEs with respect to the other systems. Nonetheless, the differences in MSE from one system to another are from minimal to insignificant so it is perhaps not a good way to differentiate the systems.

On the other hand, we can make a more detailed assessment by visual inspection of the results. We present the results for $\sigma = 22.5$ in Figure 67 and Figure 68. For comparison purposes we also include the denoising result using the `wiener2` function from MATLAB. For the UDWT we only show the 9-7 filters result.

First we see the superior quality of wavelet thresholding methods compared with the adaptive noise-removal filtering implemented by `wiener2`. Except for the local statistics, Wiener filtering has no built-in means to capture accurately edge location.

Comparing the UDWT with the directional pyramids (DPs), it is clear that the UDWT shows significant blurring and ringing around the edges, while the SPyr and OCR-UDFB show well preserved edge detail. For instance, the brim on the hat is well delineated in the DPs. The OCR-UDFB shows slightly better edge sharpness and less noise artifacts than the SPyr.

On the other hand, the noise artifacts that remain on the DPs are more apparent over

(a) Original

(b) Noisy

(c) Wiener

Figure 67: Denoising results using Lenna contaminated with AWGN with $\sigma = 22.5$.

smooth regions. These artifacts present directional structure resembling brush strokes. The SPyr seems to have stronger artifacts. The UDWT seems to be pretty good at removing noise over smooth regions, and the remaining blips are less objectionable to the eye.

5.6 Conclusions

In this chapter we have explored the application of UDFB-based pyramids in the problem of non-linear image denoising using subband domain thresholding. Since our decomposition provides better directional selectivity, we were expecting to have better representation of edge detail at the subband level.

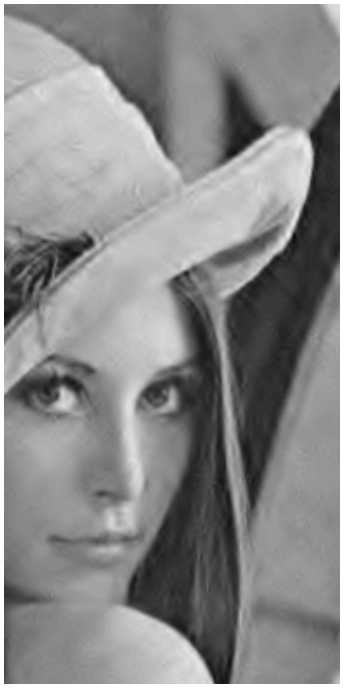
From our experiments, we showed that from the MSE point of view, increased directional selectivity has impact. From the perceptual point of view, we did see some improvement in the retention of edge sharpness over the UDWT case. However, the cross-hatching artifacts encountered in smooth regions could be more objectionable than the isolated blips in the UDWT.

Between the OCR-UDFB and the SPyr, the OCR-UDFB seems to have less artifact distortion and consistently provides better MSE. So we could say that the OCR-UDFB is slightly favored based on the experiments performed for this thesis. However, the differences are not significant enough to identify a clear winner.

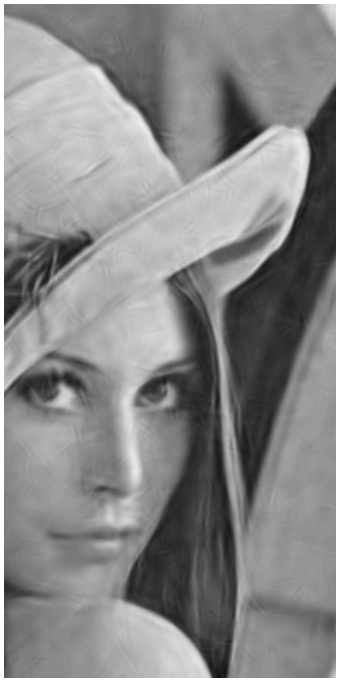
Future work should concentrate in deriving some hybrid scheme in which the best features of the OCR-UDFB and the UDWT such that sharp edge detail is preserved while smooth regions have minimal noise artifacts.

In the next chapter we apply the denoising schemes describe here to the problem of speckle removal on SAR images.

(a) SPyr



(b) UDWT



(c) OCR-UDFB

Figure 68: Denoising results using Lenna contaminated with AWGN with $\sigma = 22.5$.

CHAPTER VI

ENHANCEMENT, ANALYSIS, AND COMPRESSION OF SAR IMAGES

Synthetic aperture radar (SAR) imaging has evolved into an important tool for the study of physical and social geography. The formation of SAR images is beyond the scope of this work, but several books cover the topic. In particular, Curlander, et al. [26] and Jakowatz [57] give a detailed treatment of SAR systems. In a nutshell, a SAR image is formed by radiating the area of interest with a coherent source, and then collecting and integrating scatterers over a synthetic (antenna) aperture.

SAR image analysis is a very active area of research. For instance, it has been used in the study of deforestation of the rain forest, crop distribution, catastrophe assessment, glacier displacement, and urban sprawl. Since the original use of SAR imaging was military in nature, a large amount of work is still done on surveillance and automatic target recognition. One main advantage of SAR over other imaging techniques, is that it can generate imagery under all weather conditions. A good review on this applications is given in a book by Oliver and Quegan [93].

In this chapter our objective is to apply the DFB-based processing and analysis tools, developed in the previous chapters, to the enhancement, analysis and compression of SAR images. First we deal with speckle removal (i.e. enhancement) using subband coefficient shrinking. We compare our results with those of the UDWT and show that the DFB provides better speckle removal. Next, we study the problem of land cover/terrain segmentation of SAR images by exploiting their textural characteristics. Finally we study the compression of SAR images using a model-based scheme.

6.1 Overview of Speckle Filtering Techniques

A major difficulty associated with the use of SAR images is the presence of speckle. Visually, speckle is a granular noise that gives the image a richly textured look. Physically, speckle is a by-product of illuminating an area with a coherent source over a finite aperture. Hence, the resulting images have a limited resolution. Speckle is formed when random scatterers within a resolution cell add together coherently. When the illuminated surface is rough with features larger than the signal wavelength, the phase of the scatterers causes random fluctuations akin to a random walk process.

As can be expected, speckle hampers the analysis of SAR images both visually and with image processing tools. Speckle can be modelled as multiplicative noise, which makes it difficult to eliminate with classic filtering schemes. A careless filtering operation could distort the underlying structure or destroy non-speckle textural information useful to analysis. Needless to say, speckle removal has received significant attention. See [82, 124, 92] for reviews on speckle-filtering methods.

The multiplicative nature of speckle is usually represented as

$$z(i,j) = x(i,j)n(i,j),$$

where n represents the speckle noise, and x is the backscatter that needs to be recovered. Some techniques have been developed to work directly with the multiplicative noise model. However, as there is a vast body of work on additive noise, a logarithmic transformation can be applied giving the expression

$$\log z(i,j) = \log x(i,j) + \log n(i,j).$$

As mentioned in [4, 52], the distribution of $\log n(i,j)$ approaches a Gaussian. On the other hand, the logarithm introduces a bias on the data which can destroy the radiometric fidelity of the data. As noted in [139] the bias should be removed before the filtered data is exponentiated back to the original domain.

In general, despeckling techniques use adaptive filters which attempt to estimate the true radar cross section by doing a local-global statistical analysis for each pixel on an

image. The most popular schemes by Lee [73], Kuan [71], and Frost [39] are solutions to the minimum mean square error(MMSE) filter for both multiplicative and additive speckle. For instance, for the additive case

$$z(i,j) = x(i,j) + n(i,j)$$

the estimate $\hat{x}(i,j)$ is given by

$$\hat{x}(i,j) = (1 - \beta(i,j))\overline{x}(i,j) + \beta(i,j)z(i,j), \tag{86}$$

where

$$\beta(i,j) = \frac{\sigma_x^2(i,j)}{\sigma_x^2(i,j) + \sigma_n}$$

The noise variance σ_n^2 is estimated from a homogeneous region in the image. The a priori mean $\overline{x}$ and variance σ_x^2 for each point (i,j) are estimated from the noisy data over a local window (commonly 7×7) and is given by

$$\sigma_x^2(i,j) = \sigma_z^2(i,j) - \sigma_n$$

and

$$\overline{x}(i,j) = \overline{z}(i,j).$$

With the introduction of wavelet-based denoising techniques, the application to speckle removal followed. Most common despeckling techniques are based on local/global statistical adaptive filters that are computationally demanding. Hence, the simple approach of thresholding is attractive. However, the nature of SAR images varies significantly from the assumptions made on the wavelet denoising problem. Namely, the noise was assumed to be Gaussian, white and additive. Hence some adjustments have to be made to wavelet schemes to work on SAR images as follows:

1. As described before, the additive assumption can be met by doing a logarithmic transformation as reported in [90, 134, 138, 1]. However, care should be taken on removing the bias introduced by the logarithm, otherwise radiometric distortion will be introduced. On the other hand, we can stay in the multiplicative domain and apply local processing schemes as reported in [40, 36].

2. Regarding the distribution of the noise, it has been shown [4, 52] that the pdf of the noise logarithm approaches a Gaussian. Hence converting to an additive noise model seems advantageous. On the other hand, it has been reported that wavelet denoising schemes are also applicable to non-Gaussian scenarios. This can be explained by the unconditional basis property of wavelets.

3. Regarding correlated noise sources, it has been shown by Johnstone and Silverman [60] that subband level thresholds are good enough. Odegard, et al., [90] used a pre-whitening step, however this introduces severe radiometric distortion. Finally, a common way to break the speckle correlation has been to downsample the image as reported by Achim, et al. [1].

Regarding the subband coefficient modification step, a few techniques have been reported that try to impose better statistical models on the data. We can identify two main categories: local and subband level schemes. Local thresholding schemes have focussed more on understanding and exploiting the spatial and directional localization provided by the DWT. The work of Fukuda and Hirosawa [40] consist of modifying the wavelet coefficients based on magnitude and local geometrical features. Pizurica, et al. [99, 100] developed schemes that can be seen as a refinement of Fukuda's work. The resulting shrinking curves are smooth and provide better visual quality. Finally, Xie et al., [138] uses a Markov model to capture local geometrical structures.

For subband level processing, recent work tends towards a Bayesian approach, where prior distribution models for the clean subband coefficients and noise are established, and a MAP classifier is obtained based on these assumptions. Zeng and Cumming, use a generalized Gaussian model for the clean coefficients and a Gaussian distribution for the speckle noise [141]. Achim, et al., [1] have reported a MAP estimator using symmetric-alpha-stable distributions for the underlying signal, and a Gaussian distribution for the log-transformed speckle. Foucher, et al. [36] have developed a generalization of the Gamma-filter [77]. A sigmoidal "shrinking curve" is the outcome of the MAP classifiers. These schemes are computationally demanding and might be suitable only for off-line applications. Moreover, as

can be seen, there is no universal model and model assumptions have to be adjusted for different types of SAR images.

6.2 Speckle reduction using the DFB

We apply the DFB-based systems developed in this thesis in a similar way to the wavelet-based schemes. As discussed in Chapter 5, the despeckling scheme consists of the three steps: 1)transform, 2)subband coefficient shrinking, 3) inverse transform.

We have selected a directional pyramid (DP) using the UDFB and the multiresolution pyramid from [118]. We conjecture that improving the orientation selectivity of the transform will provide a better isolation of edges and structures. Hence, we focus on SAR images that contain a good amount of man-made structures like buildings, roads, vehicles, etc. The resulting images are compared against the UDWT.

The frequency domain partitioning of the DP is shown in Figure 69. The radial component has a residual high frequency component that captures frequency content for $\sqrt{\omega_0^2 + \omega_1^2} > \pi$ which is not directionally decomposed. The mid-frequency bands are decomposed with an eight-band UDFB. We made two considerations for selecting this DP configuration. First we want to keep complexity similar to that of the UDWT; a fully undecimated DP (radial- and directional-wise) would require significant more storage and computation making. Second, we use the UDFB instead of the maximally decimated DFB in order to retain shift invariance at each resolution level.

For the thresholding algorithm, we initially considered the algorithms from Fukuda and Hirosawa [40], and Pizurica, et al. [99]. Our initial evaluations showed that Fukuda's algorithm tended to leave too much texture, and the resulting image was perceived as less sharp than those processed with Pizurica's algorithm. Therefore, we chose the Pizurica algorithm for our subsequent work.

We present pseudo code for the Pizurica algorithm in Figure 70. The threshold for each subband coefficient is determined by analyzing the magnitude of the coefficient and the structure of a surrounding local neighborhood. The local neighborhood aligns with the orientation of each subband. In Figure 71 we contrast the neighborhood definitions for the

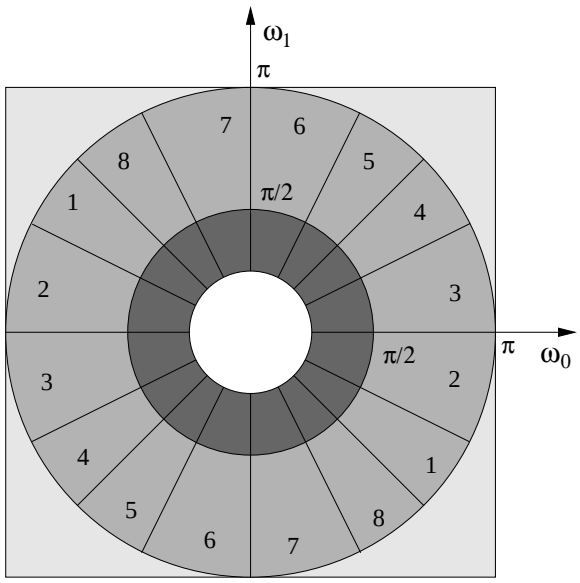


Figure 69: Frequency plane partitioning for the radial-UDFB pyramid with eight directional bands.

UDWT and UDFB for this algorithm. Obviously, there are three different neighborhoods for the UDWT and eight for our UDFB system.

The shrinkage factor q_ℓ depends on two terms. First, ξ_ℓ determines the impact of noise on a coefficient. Let $\alpha = 1$ so that

$$\xi_\ell = \frac{m_\ell}{2T - m_\ell}.$$

We can see the behavior of these term at some points. If m_ℓ is considerably lower than T, then $\xi_\ell \to 0$. If $\xi_\ell = T$, then $m_\ell = 1$ and this term is neutralized. Obviously, anything greater than T will indicate the coefficient belongs to the signal component.

The second term, η_ℓ measures the impact of the local neighborhood. As is well known, coefficients of large magnitude belonging to edges tend to cluster together. Hence, a coefficient surrounded by significant coefficients will have a higher probability of being significant too. On the other hand, an isolated large coefficient, will most probably be an outlier that would manifest as a blip in the denoised image. To take into account the spatial geometric constraint, the structure around a pixel is measured by

$$t_\ell = \sum_{k \epsilon N_{\theta,\ell}} (2x_k - 1)$$

function PizuricaShrink(w, N_θ, α, β, $gamma$ w^{new})
    Input: w ▷ Noisy subband
    Input: N_θ ▷ Directional neighborhood
    Input: α ▷ Used to define threshold ranges
    Input: β ▷ Controls impact of noise measure ξ_ℓ
    Input: γ ▷ Controls impact of spatial constraint η_ℓ
    Output: w^{new} ▷ Cleaned subband

    $m = |w|$ ▷ Noise measure
    $T = MAD(w)/0.6745$ ▷ Global threshold
    $x = m > T$? 1:0 ▷ Binary mask image
    for all subband coefficients w_ℓ **do**
        Calculate magnitude dependent noise measure

$$\xi_\ell(m_\ell) = \frac{m_\ell - (1-\alpha)T}{(1+\alpha)T - m_\ell}, \quad (1-\alpha)T < m_\ell < (1+\alpha)T$$

        Calculate neighborhood activity

$$t_\ell = \sum_{k \epsilon N_{\theta,\ell}} (2x_k - 1)$$

        Calculate local spatial constraint parameter

$$\eta_\ell = \exp(t_\ell)$$

        Calculate the Shrinkage factor

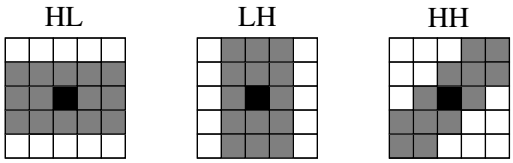
$$q_\ell = \begin{cases} 0, & m_\ell \leq (1-\alpha)T \\ \frac{\xi_\ell^\beta \eta_\ell^\gamma}{1+\xi_\ell^\beta \eta_\ell^\gamma}, & (1-\alpha)T < m_\ell < (1+\alpha)T \\ 1, & m_\ell \geq (1+\alpha)T \end{cases}$$

        Apply shrinking factor

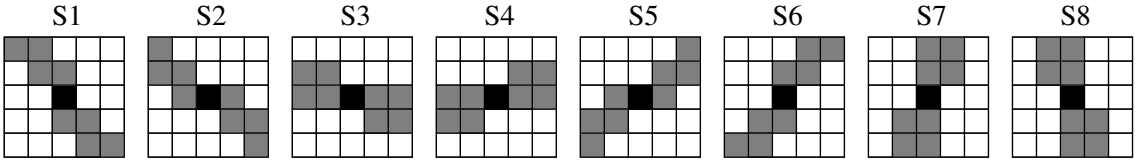
$$w_\ell^{new} = w_\ell \cdot q_\ell$$

    end for
end function

Figure 70: Pseudo-code for Pizurica subband despeckling algorithm



(a) UDWT Neighborhoods



(b) Eight−band UDFB Neighborhoods

Figure 71: Definition of local neighborhoods for the Pizurica algorithm. The center pixel (black) is the pixel to be thresholded. The gray pixels form the neighborhood N_ℓ for each subband orientation.

where x_k are binary values. A value of $t_\ell > 0$ indicates that most of the pixels in N_ℓ are considered significant. Otherwise, the neighborhood is considered as having no local structure. To measure the geometrical constraint we use

$$\eta_\ell = \exp(t_\ell),$$

and note that for $t_\ell = 0$, the effect of this term is neutralized.

Hence, the product $\xi_\ell^\beta \eta_\ell^\gamma$ in q_ℓ increases (decreases) when both terms increase (decrease) simultaneously. When terms diverge, then all sorts of possible situations arise. For instance, there will be cases where a coefficient magnitude is small, but is located close to a "significant" cluster, producing a q_ℓ close to one. Hence the algorithm generates a smooth thresholding curve of sigmoidal shape as shown in Figure 73. This provides a gradual transition from speckle to structure coefficients.

6.2.1 SAR despeckling results

For evaluation, we use the "horse track" image obtained from the Sandia Labs website [115]. The image is presented in Figure 72. The image clearly presents linear and curvilinear structures at different orientations. This is magnitude image that has been log transformed, and re-scaled to 256 grayscales. Hence, for practical purposes, the speckle noise is assumed

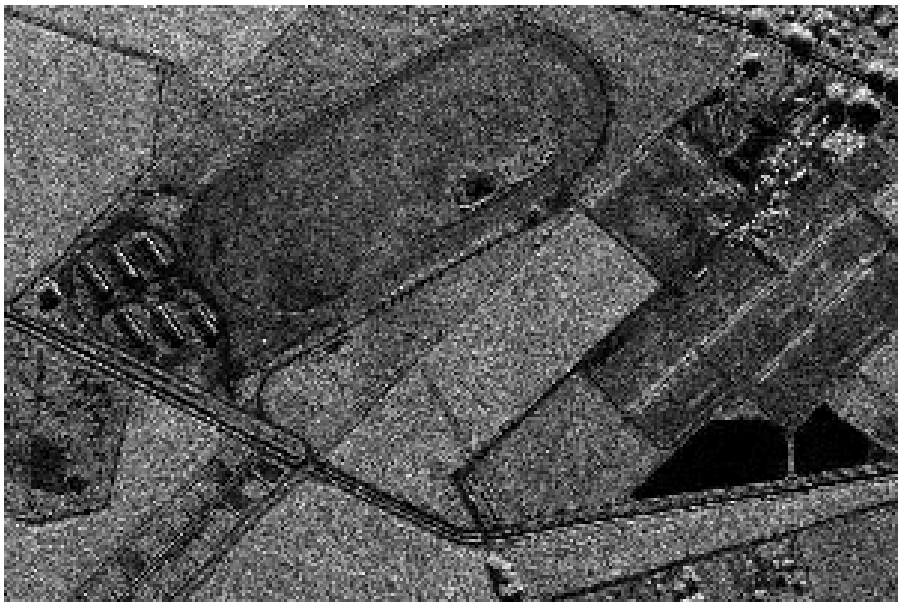


Figure 72: Original horse track image.

additive.

For the UDWT we used a transform with two pyramid levels using the 9-7 biorthogonal filters. For the directional pyramid (DP), we calculated four pyramid levels (including the residual and lowpass components) and an eight-band UDFB on the mid-band channels. The UDFB was implemented with three-stage ladder structures using $\beta(z)$ filters of length eight which were designed with the maximally-flat method. The number of pyramid levels was determined based on previous work [117, 113] which has determined that speckle information resided at spatial frequencies $\sqrt{\omega_0^2 + \omega_1^2} > \frac{\pi}{4}$.

For the shrinking algorithm, we have selected empirically the values $\alpha = 1$, $\beta = 2$, $\gamma = 2$. The algorithm produce a shrinking curve that depends on the value of T which is a subband-level threshold. For subband S_i, we calculate T as $T = MAD(S_i)/.6745$, i.e., the robust standard deviation estimate. An example of a shrinking curve is presented in Figure 73.

In order to evaluate the results we need some type of metric. As reviewed in [82], objective measures like the Equivalent Number of Looks (ENL), have been proposed, but

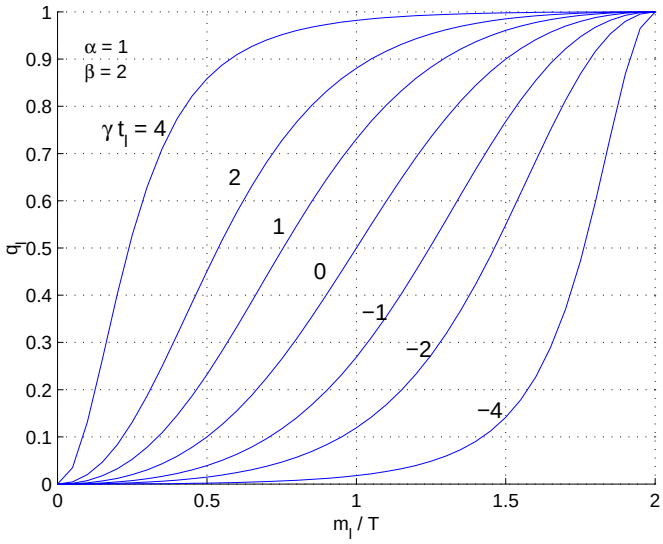
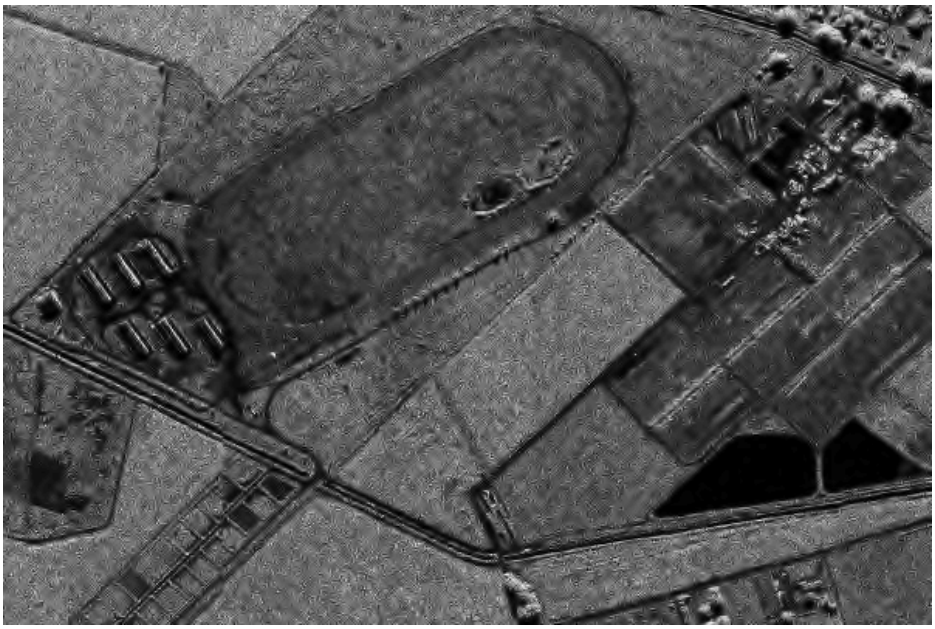


Figure 73: Example of a family of shrinking curves generated by the Pizurica algorithm (Algortihm 70). In this case we have $\alpha = 1$,$\beta = 2$, and the product $-4 \leq \gamma t_\ell \leq 4$. The abscissa is the value m_ℓ normalized by T.

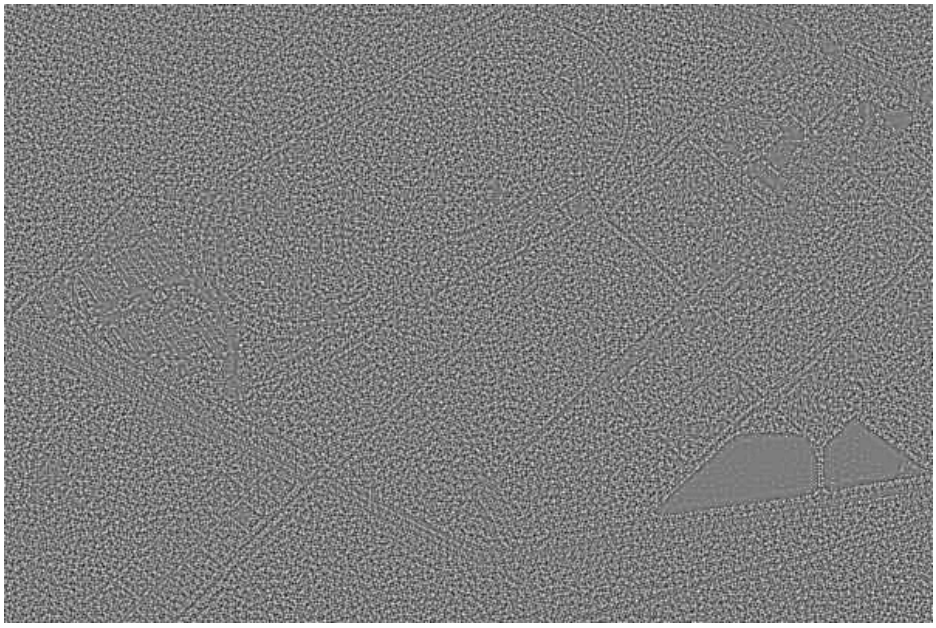
they are only useful to evaluate homogeneous areas. These areas have to be handpicked for each image (i.e., the measure is not global). As we are trying to evaluate edge retention over the whole image, these objective measures do not work. The best way to assess the performance of the algorithm is by doing side to side visual comparisons. In order to appreciate the separation of structure detail from speckle comparing residuals provides good insight. In fact, use of residual has been used to subjectively measure the quality of speckle reduction in [93, 124].

For the horse track image we have significant edge detail oriented in multiple directions, as well have curvilinear structures. The results are presented in Figures 74, 75 and 76. First, we see that the radial-UDFB shows less "noise blips" than the UDWT system. From the residuals we see that the UDWT looses a significant amount of edge information while the edge detail in the radial-UDFB residual is much more difficult to perceive. For instance, we can see in the residual of Figure 74 that the outline of the horse track and the roads are much stronger than in the residual of Figure 75.

We observed similar results in other test images, where the residuals from the UDWT show clear loss of edge detail, while the DP-based system presents minimal to imperceptible

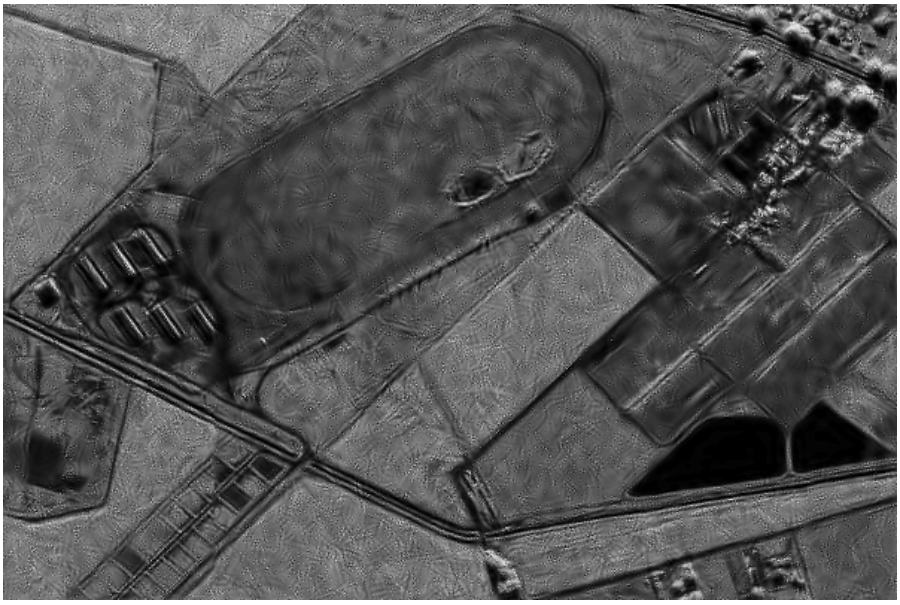


(a) Despeckling result

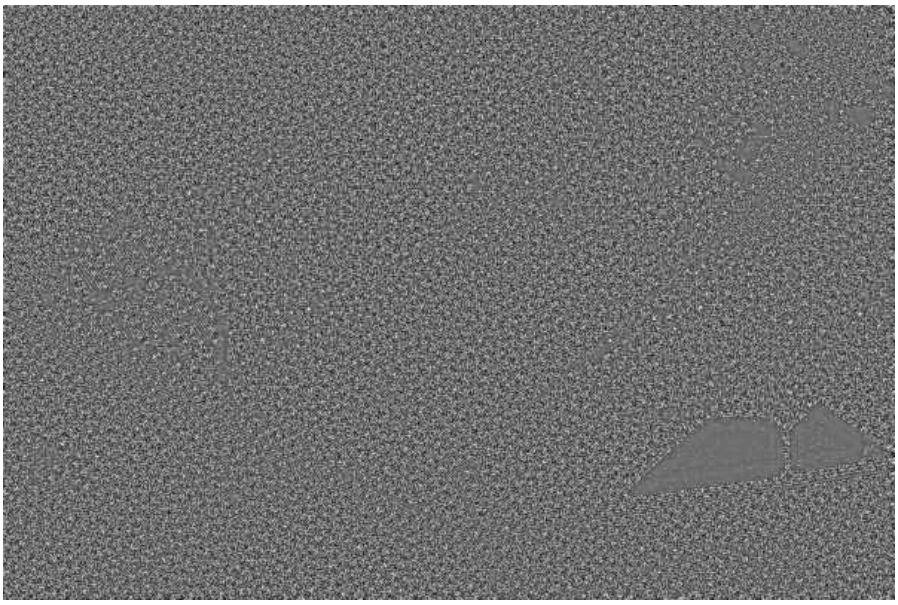


(b) Residual

Figure 74: Despeckling results using UDWT.



(a) Despeckling result



(b) Residual

Figure 75: Despeckling results using directional pyramid with the UDFB.

edge details. Opposite to the results in Chapter 5, we found that for SAR images, the denoising algorithms based on the UDFB have much better edge information retention than the UDWT. Later in this chapter, we use this despeckling scheme to implement a model-based SAR compression algorithm.

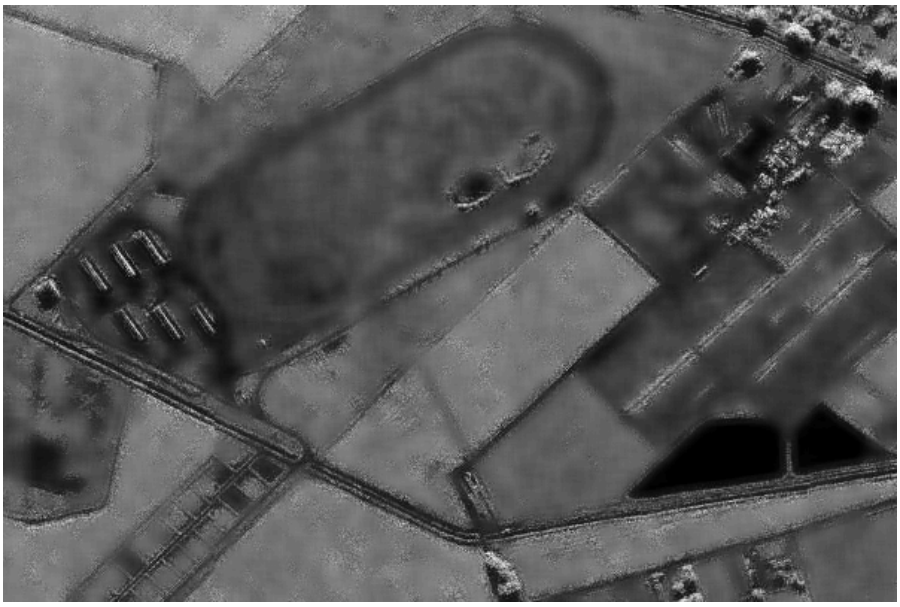
Finally, we compare the despeckling results with the results from the MMSE filter from Lee described in the previous section. Previous comparisons between local statistic filters and wavelet-based schemes have been inconclusive [50, 42]. We show the despeckled image and the residual in Figure 76. As seen, this filter has good speckle removal properties over the smooth regions, however it has a big problem separating edge and structural information from speckle as can be seen in the residual image, and the significant loss of detail on the filtered image. For images where structural features are an important component, the UDWT and DP methods described in this section should be the preferred choice.

6.3 SAR image segmentation using the DFB

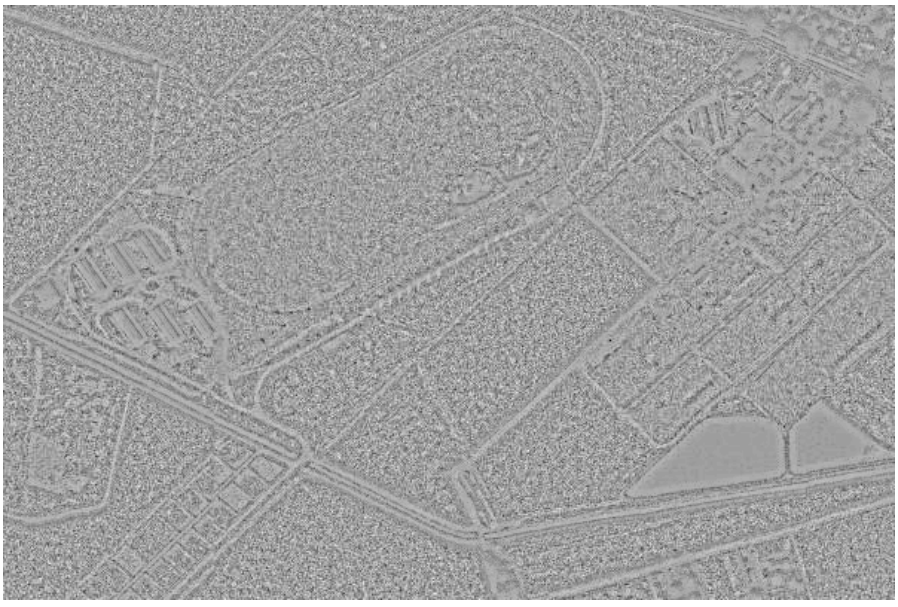
An important application of SAR image analysis is the study of vegetation and crops distributions. In SAR images different types of land cover are identified by the texture attribute of the image. As expressed by Ulaby, et al. [125]: *"Texture may in fact be more useful than image tone in interpreting radar images."* Hence, land cover classification and segmentation algorithms based on texture have been reported in several works, see [125, 104, 136] for examples.

We will apply the texture classification and segmentation algorithms developed in Chapters 3 and 4 to SAR images. We start by selecting test images from the MSTAR clutter public database [89] which were captured during a flyover of a Maryland rural area. These SAR images are in the magnitude-phase format (only the magnitude is used). The images cover fields, woods, and sparse settlements. We present three test samples (HB06226, HB06246, HB06186) in Figure 77.

Our objective will be to segment each of these images into field, tree and shadow regions. Identifying shadows is useful for surveillance operations, as these areas can be used for hideouts. We obtain 256×256 training patches for these regions from a different set of

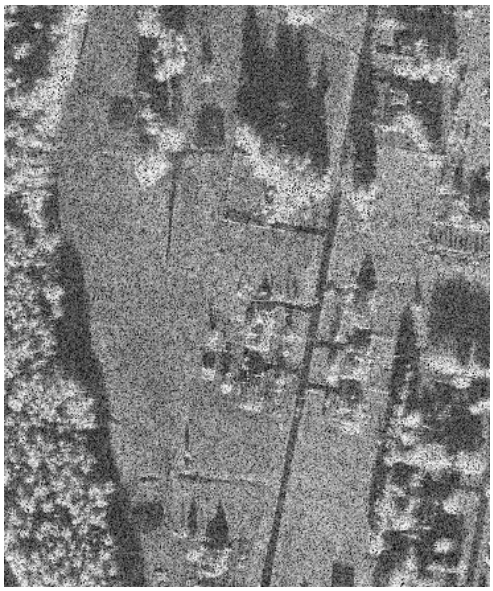


(a) Despeckling result

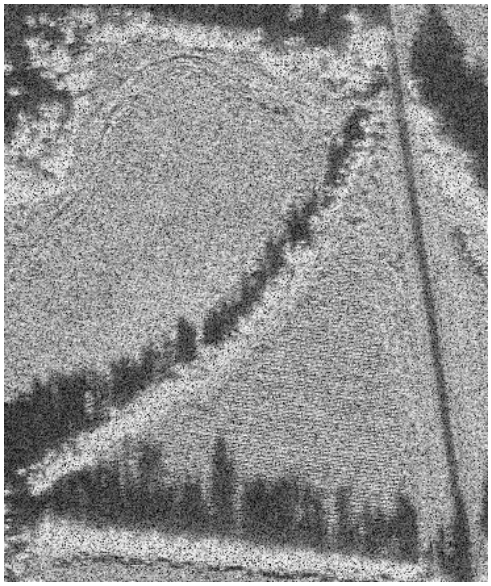


(b) Residual

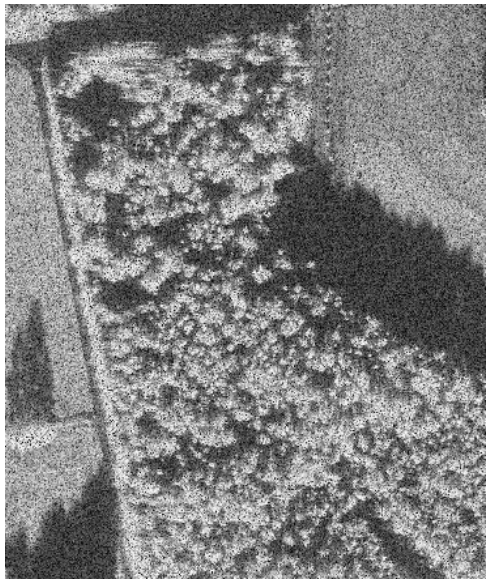
Figure 76: Despeckling results using Lee's MMSE filter.



(a)HB06226



(b) HB06246



(c) HB06186

Figure 77: MSTAR database test images.

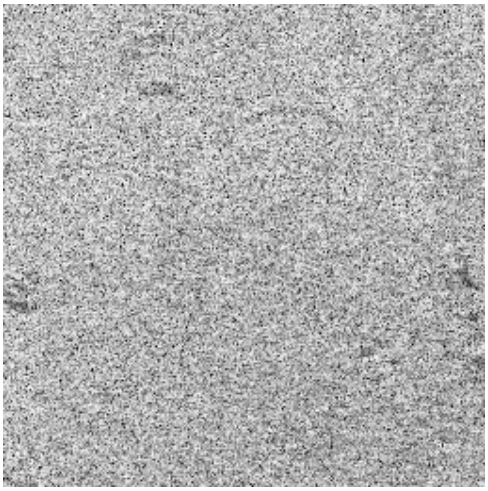
images (HB06173, HB06267, HB06238, HB06188). The training samples are presented in Figure 78. As will be explained later we extracted two different types fields. We note that visually *fields1*, *fields2* and *shadows* textures are differentiated by the average grayscale value. However, as our segmentation algorithms rely on mid-frequency bands only, there are indeed textural differences on the samples.

We used both segmentation algorithms from Chapter 4. These algorithms differ in the directional pyramid used. In this case, we use the UR-UDFB and UR-DFB pyramids introduced in Chapter 5 with $J = 5$ and $N = 8$. The radial decomposition is the same used in the steerable pyramid [118] which produces as additional residual band containing the frequency the frequency information at $\sqrt{\omega_0^2 + \omega_1^2} > \pi$. As speckle resides in the highest frequency bands, we found that the highest frequency band provided no information useful for segmentation. In fact, the segmentation quality was poorer when this band was included. For the DFB and UDFB, we used a three-stage ladder structure with step maximally-flat filters with $L = 8$. Hence, the feature set consist of 24 features. For the classifier, the LVQ algorithm uses five codewords per class.

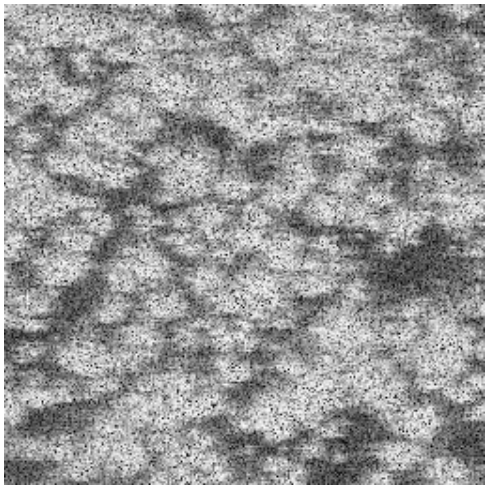
For HB06186 and HB06246 we used the training patches for trees, shadows, and fields1. We present the results in Figures 80 and 81 respectively for both directional pyramids. We see good identification of the three zones. In H06246 we see that the thin diagonal tree line is delineated, as well as the road which is classifies as a shadow. Hence our feature set has very good spatial discriminatory power.

For H06186, the upper-right region is misclassified as shadows in the UR-UDFB system. Looking closely at the image, we see that in terms of average grayscale, this region looks darker than the rest of the field cover. As we are not using the lowpass subband for feature extraction, then we infer that what is perceived as a tonal difference is caused by having a different texture in that area which the classifier identifies as being closer to the shadow class. For the UR-DFB system, this misclassification is not present.

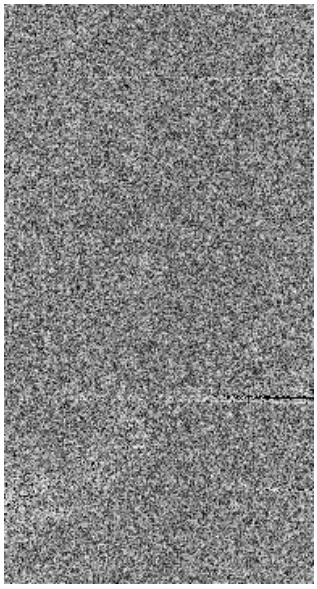
For region HB06226, we found that the field regions where significantly misclassified as shadows. This implied that the textural characteristics of the fields in HB06226 were



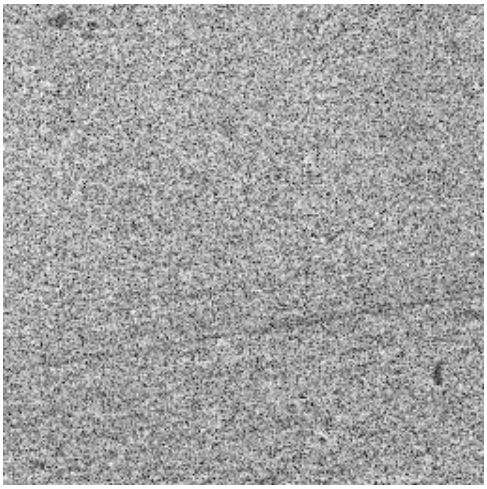
(a)HB06173-fields1



(b)HB06267-trees

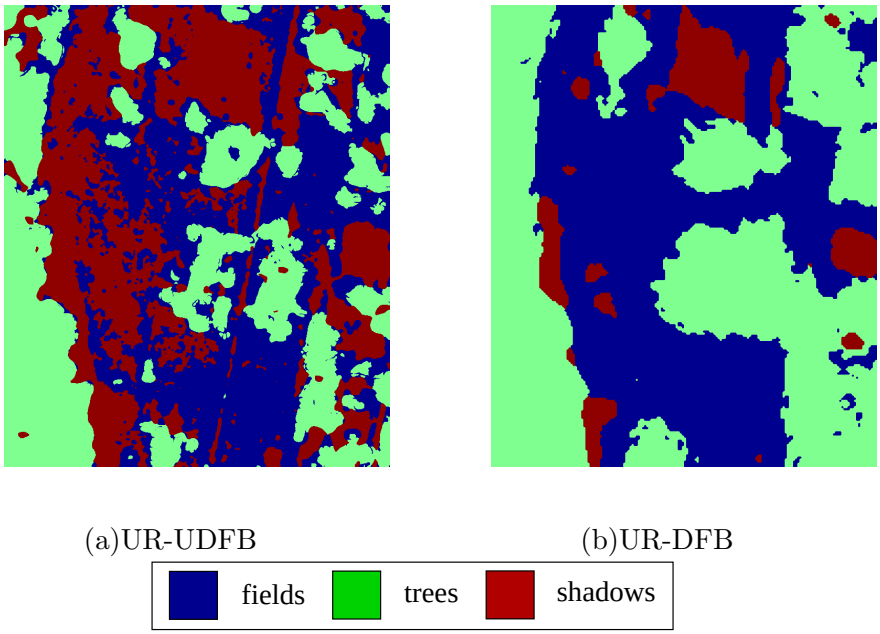


(a)HB06238-shadows



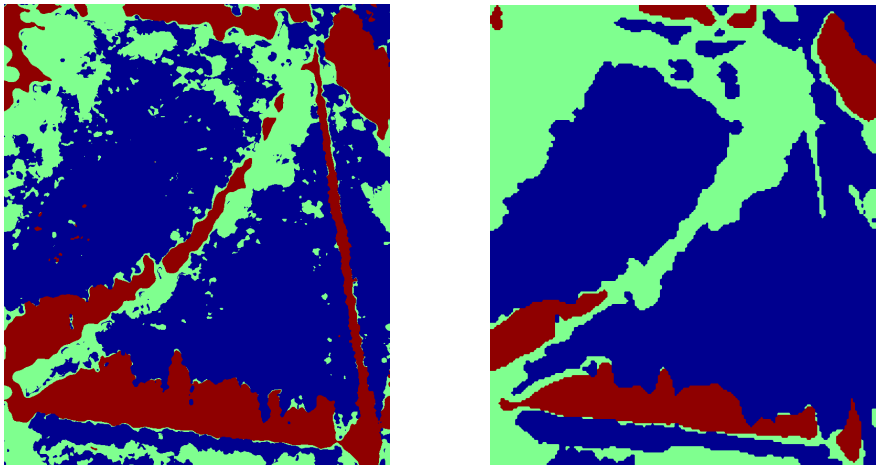
(b) HB06188-fields2

Figure 78: Training images/textures extracted from MSTAR images.



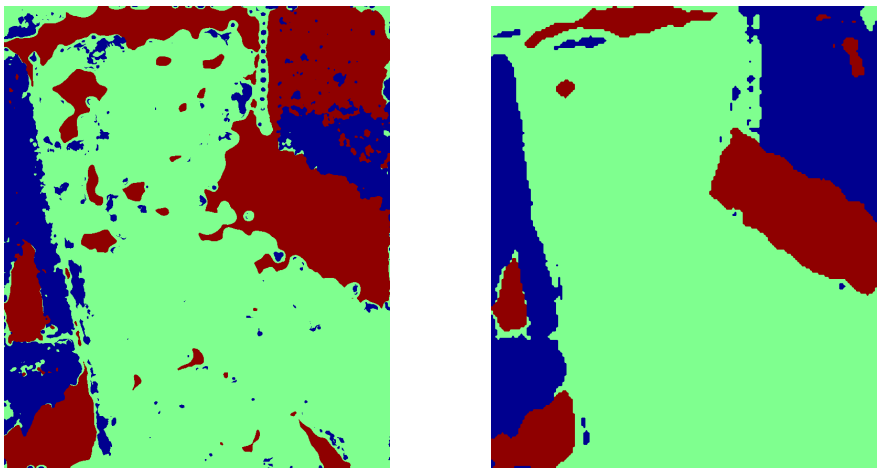
(a)UR-UDFB (b)UR-DFB

Figure 79: Segmentation results for image HB06226.



(a)UR-UDFB (b)UR-DFB

Figure 80: Segmentation results for image HB06246.



(a)UR-UDFB (b)UR-DFB

Figure 81: Segmentation results for image HB06186.

not the same as in the training patch. This could mean that a different type of vegetation was covering this area. We substituted the fields patch with a region extracted from HB06188 (fields2). The segmentation results are shown in Figure 79. Although we see a good delineation of most areas, there is still erroneous labelling of field cover as shadow. This indicates that our selection of training data does not completely capture the land cover characteristics of this image. It makes it apparent that we need to have some information of the ground truth, which in many cases is difficult to obtain. Hence, some geographical side information of the imaged region would be useful in order to choose an optimal set of land cover classes.

These results also show a limitation of our segmentation algorithms, the building structures in the image are classified as wooded regions. We have not included a "buildings class", as these are non-homogenous areas difficult to model as textures. Hence, these segmentation algorithms should be used as a frontend of an analysis system, where the frontend subsystem identifies the overall distribution of land covers. Another subsystem would then take segmentation results and do further analysis of the image.

Comparing the segmentation results, we see that the DFB-based results present good

level of recognition of large areas, however the definition of the region boundaries is poor compared to the UDFB-based system, and small patches are not resolved well. On the other hand, the DFB system provides eight-fold savings in computation and storage requirements when compared to the UDFB case. Ultimately, we have the flexibility of choosing a system depending on the resources available and/or the segmentation quality desired.

6.4 Compression of SAR Images

SAR images have inherently different statistical properties from natural images. Thus, they do not lend themselves to efficient compression using algorithms developed primarily for natural images. How best to code SAR remains an open question. For instance in [97], several options are suggested including compressing the telemetric data, using the complex valued image, and dealing with the intensity or magnitude products. Several approaches to compression have been considered in [16, 9, 34, 114, 94, 134]. Rosiles, et al. [43] presented an evaluation of different compression algorithms for SAR images including JPEG, wavelet codecs, peceptual- and model-based compression, and vector quantization schemes. Overall, it is recognized that the huge amounts of imagery produced by SAR systems, requires development of custom compression techniques.

Also, the intended use of the images has to be kept in mind. For image analysis applications, it would be desirable to keep the radiometric information as unaltered as possible. For human analysis, the perceptual characteristics should be retained and structures should remain unaltered. The major difficulty in achieving good compression with high quality is speckle. Because of the high frequency characteristics of speckle, the entropy of SAR images is high. Hence, state-of-the-art image coders for natural images are at not well fitted for SAR images.

In this research, we consider the application where the images are used for human visual inspection. Image analysts are experts trained to interpret the contents of SAR images usually for military purposes. Analysts prefer the image data to remain as untouched as possible. Although speckle is not a conveyer of relevant information, it can influence the decisions of the image analyst.

In this thesis, we introduce model-based technique that recognizes two components in a SAR image: the underlying scene and the speckle. Each component is then compressed as a natural image and modelled as a texture respectively. Previous work on this technique has been reported by Rosiles and Smith in [112, 113]. Here we present improvements in the model in terms of component separation and texture modelling as will be described next.

6.4.1 Image-Structure / Speckle-Texture Model for visually lossless SAR image compression

We consider an image model perspective in which the speckle is viewed as a texture and an approximation to the infinite resolution version of the SAR is extracted from the input. The image model in this case views a SAR image $X(i,j)$ as the combination of two components: an image structure $S(i,j)$ and as speckle texture $T(i,j)$, the form of which is given by

$$X(i,j) = \mathcal{R}(S(i,j), T(i,j)),$$

where $\mathcal{R}()$ is a recombination rule for the two components [31].

The encoding process of the algorithm involves in the first step extracting $S(i,j)$ and $T(i,j)$ and then separately coding the components; $S(i,j)$ is coded using a conventional subband image coder, while $T(i,j)$ is analyzed and parameterized as a stochastic texture. A block diagram of these scheme is presented in Figure 82. The decoder stage consists of decoding $S(i,j)$ and synthesizing $T(i,j)$. A final postprocessing step takes into account the shadows in a SAR images by scaling down the values of $T(i,j)$ in these areas. The shadows are found by a simple segmentation of $S(i,j)$ as described later.

We propose a structure-texture separation scheme based on image despeckling techniques of section 6.2.1. We should note that in previous work, the DWT [112] and the DFB [113] have been used in combination with subband shrinking. However in those cases a threshold was derived by trail and error, and was fixed for each subband. Here, we use a radial-UDFB decomposition and an automatic thresholding algorithm that spatially adapts the threshold for each subband coefficient.

Once $S(i,j)$ is obtained, the texture component (i.e., speckle) is obtained by

$$T(i,j) = X(i,j) - S(i,j).$$

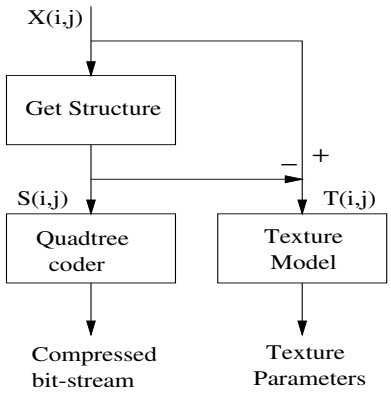


Figure 82: Block diagram for the proposed model-based encoder.

The recombination rule $\mathcal{R}()$ is then given by

$$X(i,j) = \begin{cases} S(i,j) + T(i,j) & \text{if M(i,j)} = 1 \\ S(i,j) & \text{otherwise} \end{cases}$$

Where $M(i,j)$ is a mask that signals the presence of shadow regions in the image. No texture component is added to the shadow regions in the image.

6.4.2 Coding the Structure Texture Component

Coding the image structure $S(i,j)$ component is straight-forward because this component image is similar to a natural image. Hence the image structure component is suitable for compression with conventional state-of-the-art image coders. In our coder we have selected to use the quadtree coder from Man, et al. [80], because it has some added benefits, like an exact bit rate control, and the possibility of error protection embedded in the coder.

6.4.3 Coding the Speckle Texture Component

Speckle is a stochastic process, which is rich in high frequencies. This characteristic makes it difficult to compress at low bit rates with conventional coders. Current state-of-the-art coders are tailored to deal with low frequency signals. The solution proposed in this work is to model the speckle as a texture field and use appropriate texture models for $T(i,j)$.

We are interested in texture synthesis algorithms that provide a compact parameterization of textures. In our coder we have tried two synthesis approaches, auto-regressive

models and the Heeger-Bergen algorithm [49] which captures the texture model from the subband histograms of pyramid decomposition. Both give acceptable synthesis of stochastic textures.

6.4.4 Texture Modeling Using a 2-D Auto-Regressive Model

In a survey by [53], causal 2-D autoregressive models were considered the top performers in terms of complexity, model size and synthesis quality. A 2-D AR model represents a value of a signal $w(n, m)$ as a linear combination of values in a neighborhood of (n, m) and a white innovations field $u(n, m)$, namely

$$w(n,m) = -\sum_{(0,0)<(k,l)} a(k,l)w(n-k,m-l) + u(n,m). \tag{87}$$

The two main issues for AR models are the selection of a neighborhood and the estimation of the parameters $a(k, l)$. In our case, experimental work showed that a non-symmetrical half plane (NSHP) region of support using the classical least-squares (LS) estimation algorithms led to a visually good model of T(i,j). A model order within the range (9,5) to (13,7) was deemed sufficient for the images we examined.

An AR(13,7) model requires storage of 85 parameters at floating point precision. Using 4 bytes per parameter, a 512×512 textural component can be coded at about .0104 bpp.

Provided that the 2-D AR model gives a good visual representation of $T(i, j)$, we can expect that the quality of the coded image will heavily rely on achieving good quality in the coding of $S(i, j)$.

6.4.5 Adapting the Heeger-Bergen Algorithm for Modeling of the Texture Component

The Heeger and Bergen is discussed in section 4.3, where the steerable pyramid is substituted by a DFB-based pyramid. In this sub-section, we use the directional pyramid used to obtain $S(i, j$ (see section 6.2.1). In fact, the set of all subband coefficients discarded by the despeckling process corresponds to speckle texture $T(i, j)$. In our previous work [112, 113], the despeckling and texture modelling used different transforms. Hence, using the same DP simplifies our implementation.

One of the limitations of the Heeger-Bergen algorithm is that the input texture has to be homogeneous (i.e., the statistical properties of the texture have to be approximately the same over the whole image). $T(i, j)$ is in general non-homogeneous, mainly due to the presence of shadowed regions. Shadows will bias the low amplitude bins of the histogram. Fortunately, shadows can be isolated easily from other regions in both the structure and texture component. It is possible then to view the components as bimodal (shadow/non-shadow), and do image segmentation under this criterion.

Since the location of shadows in both components is highly correlated, we perform the segmentation on the structure component. The reason for using the structure component is that texture segmentation is usually a computationally more expensive and complex procedure. On the other hand the segmentation of the image structure component (which is available at both the encoder and decoder) can be easily done by local mean classification.

From the segmentation, we generate a mask that is used to separate the texture and shadow samples. Heeger-Bergen's algorithm in Figure 62 is modified by substituting the *MatchHistogram(.,.)* function by the function *MatchMaskedHistogram(Aband$_i$, Sband$_i$, mask, level)* . Basically this function uses only the pixels indicated by the mask in calculating the histogram from that analysis band. Since subbands vary in size across resolutions, the mask is downsampled to the right size.

6.4.6 Histogram Modeling

For this application, we use the Radial-UDFB pyramid from section 5.4, in order to align the texture synthesis algorithm with the speckle removal algorithm from section 6.2. This will allow us to despeckle and calculate the subband histograms simultaneously.

The directional pyramid is an overcomplete system that generates 18 subbands. Including the input sample, the parametrization of the texture consists of 19 histograms. In our implementation, each histogram contains 256 bins plus some side information to specify its range. Therefore, the number of parameters is rather large for the purposes of transmitting at low rates (i.e. less than 1 bpp). Reducing the number of bins could help, but determining the minimum number of histogram bins so that the visual properties of the texture are

captured is a rather intractable approach. Moreover, the size of the parameter set could still be large.

In our scheme we parameterize the histograms by modelling them as a generalized Gaussian (GG) distribution if unimodal. In practice we found that most histograms exhibit a GG distribution. The normalized GG distribution is defined by

$$f(x) = a \exp(-|b(x-m)|^c), \tag{88}$$

where

$$a = \frac{bc}{2\Gamma(\frac{1}{c})}$$

and

$$b = \frac{1}{\sigma}\sqrt{\frac{\Gamma(\frac{3}{c})}{\Gamma(\frac{1}{c})}}.$$

The distribution is described completely in terms of the variance σ^2, the mean m and the power factor c; the first of these parameters determines the degree of concavity/convexity of the function. It is very easy to see that for $c = 2$ and $c = 1$ we obtain the Gaussian and Laplacian distributions respectively.

A simple technique [61, 62] exists for estimating c using the first and second absolute moments which in turn can be estimated from the data. However, the resulting parameter estimates $\hat{a}, \hat{b}, \hat{c}$ produce deficient curve fittings.

To improve the fitting of the GG distribution to the histogram, we performed the following curve fitting non-linear least squares optimization over the a, b, c parameters. The minimization problem is described by the equation

$$\min_{\{a,b,c\}} (\Sigma_{i=1}^{256}(f(x_i) - y_i)^2),$$

where the estimates of $\hat{a}, \hat{b}, \hat{c}$ are used as the initial guess in the minimization, and the sequences $\{x_i\}$ and $\{y_i\}$ correspond to the histogram's bin location and normalized bin count respectively; $f(x)$ is given by equation 88. The fitting results from this minimization routine are shown in Figure 83 for a sample of texture speckle. For the case of multimodal histograms, the curve fitting algorithm generates a GG curve with $c \geq 4$. In practice we

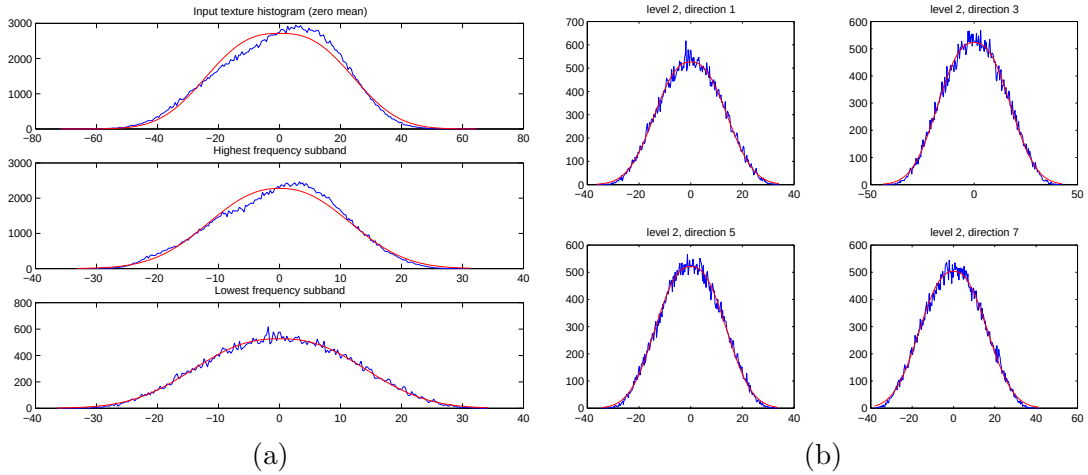


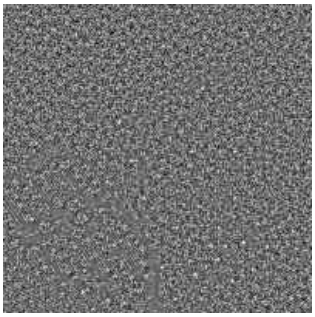
Figure 83: Example of Generalized Gaussian fitting to the histograms of the input texture and some subbands from the directional pyramid. (a)Input texture, highpass residual and lowpass subband. (b) Subbands from the second resolution level of the pyramid.

found that the synthesized texture using GG distributions only does not result in a visually distinguishable difference from the textures synthesized with the original data histograms.

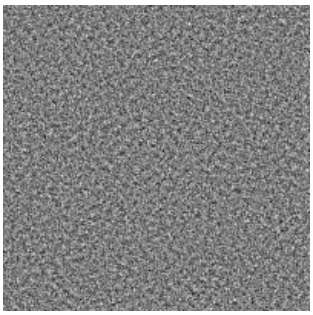
In Figure 84 we show the original texture sample, and two synthesized textures—one with the actual histograms obtained from the texture, and the other from histograms obtained from the modeling of each subband histogram as a GG. Clearly there is no visual difference between both synthesized textures, while the resemblance to the original is reasonably good. The texture is the speckle component from the Urban Scene image.

Each histogram can be represented with a, b, c, x_1, x_{256} where the last two parameters represent the end-points of the histogram. Therefore for the steerable pyramid (which requires the histograms of the input texture and 18 subbands), we have a total of 95 parameters to describe the texture. We found that the byte allocation per parameter described in Table 33, was more than enough for our purposes. In particular, c is quantized to increments of .1, giving us a range of values from 0 to 25.5, clearly more than enough. Hence, each histogram requires 11 bytes, for a total of 219 bytes.

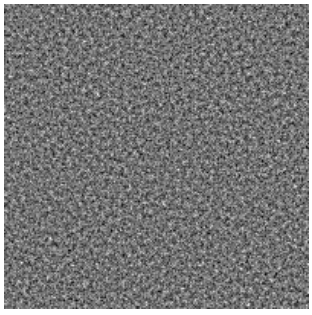
For the SAR images considered in this work, which were 8 bit 512 images, the texture parameterization requires about .007 bpp.



(a)Original



(b) AR model



(c) Heeger-Bergen model

Figure 84: Examples of texture synthesis with the two texture models

Table 33: Byte allocation for each parameter of histogram model

Parameter	Number of bytes
a	2
b	4
c	1
x_1	2
x_{256}	2

6.4.7 Compression Results

We present our best compression results for a SAR image coded at .25 bpp using the procedure just described. We have included here some preliminary results to show the dramatic improvement that is possible. As a reference we also present the compression at the same bit rates using the well known SPIHT algorithm. Perceptually we see that the SPIHT result degrades equally the textured areas and the edge detail. On the other hand, the structure component preserves better edges and structures as the compression algorithm allocates more bits to them rather than wasting the bit budget on coding the speckle. Once the texture (synthesized speckle) is recombined with the structure we have an image that is perceptually close to the original. Hence, with this approach we can give to the analyst the choice of looking at images with and without speckle while significantly reducing the storage and transmission bandwidth requirements.

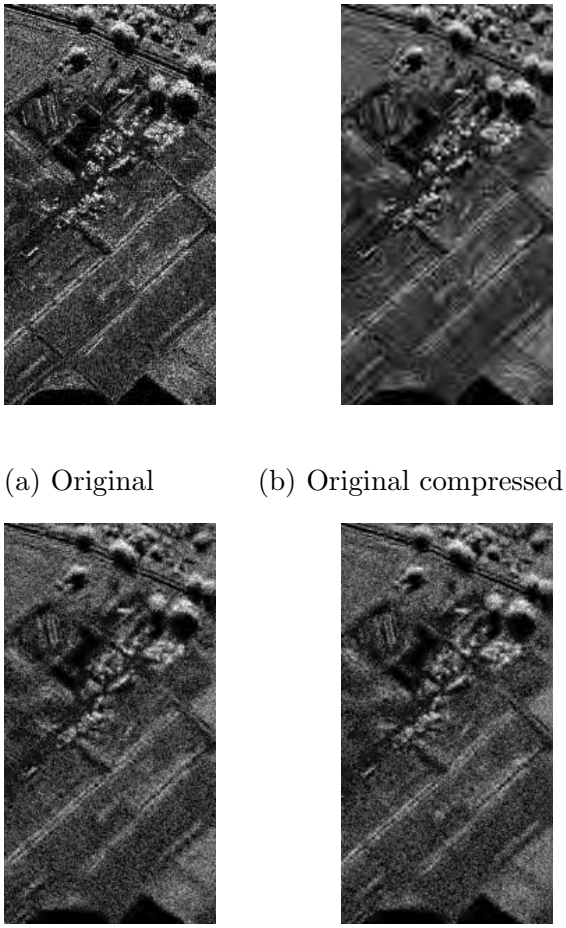
6.5 Chapter Summary

In this chapter we have applied algorithms developed in the previous chapter to the processing and analysis of SAR images. First we showed that despeckling algorithms using a radial-UDFB pyramid, retained better edge detail when compared to similar algorithms using the UDWT, and a popular local statistics filter.

Next, we adapted the texture segmentation algorithms from Chapter 4 to the problem of land cover classification and segmentation. The results show that we can differentiate field, shadow and wood regions with accuracy.

Finally, we implemented a model-based compression algorithm which provides good visual reconstruction at very low bit rates. The algorithm uses the despeckling technique developed in this chapter, and a version of the Heeger-Bergen texture synthesis algorithm which adapts the radial-UDFB pyramid.

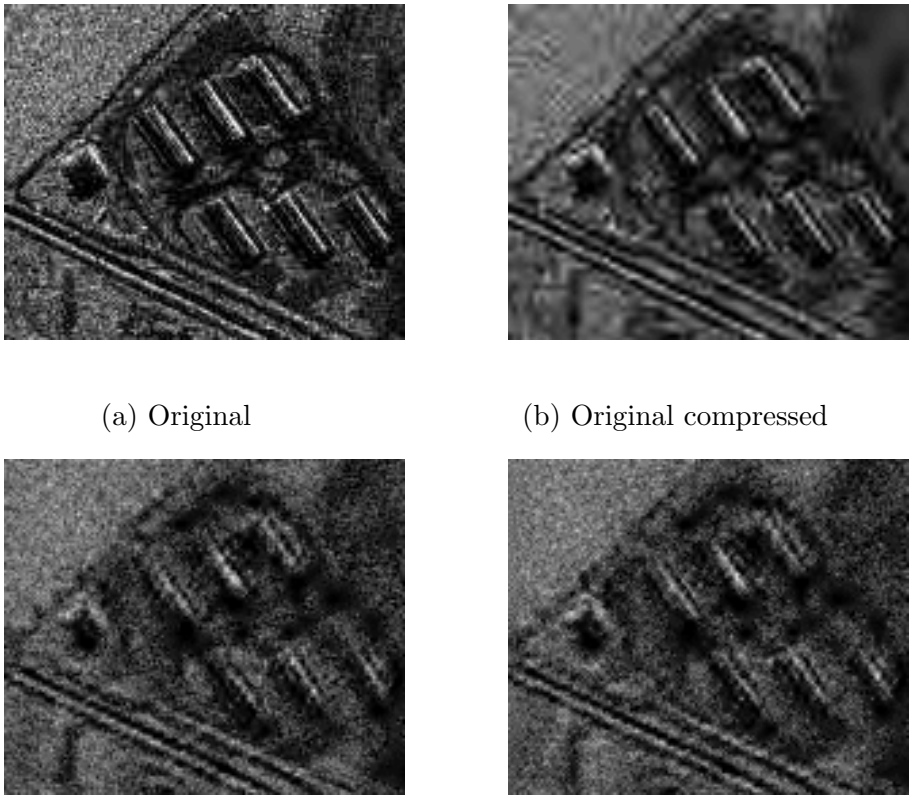
This chapter is mostly exploratory in nature. There are many possible paths to take, not only for SAR images but also for other imaging technologies like ultrasound and multispectral. The work of this chapter shows that algorithms based on the DFB and UDFB are well suited to tackle image analysis problems.



(a) Original (b) Original compressed

(c) AR model (d) Heeger-Bergen model

Figure 85: Detail from horse track image at 0.25 bpp



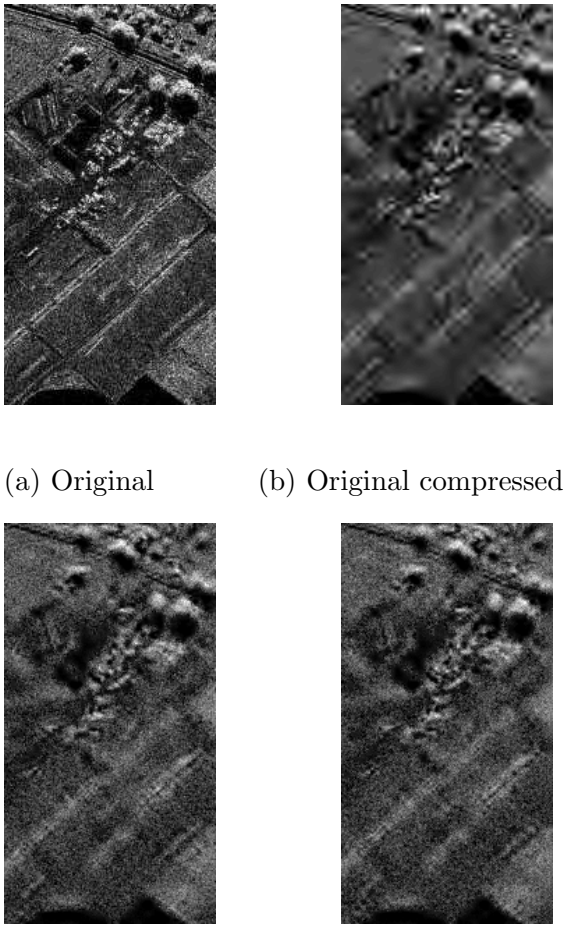
(a) Original

(b) Original compressed

(c) AR model

(d) Heeger-Bergen model

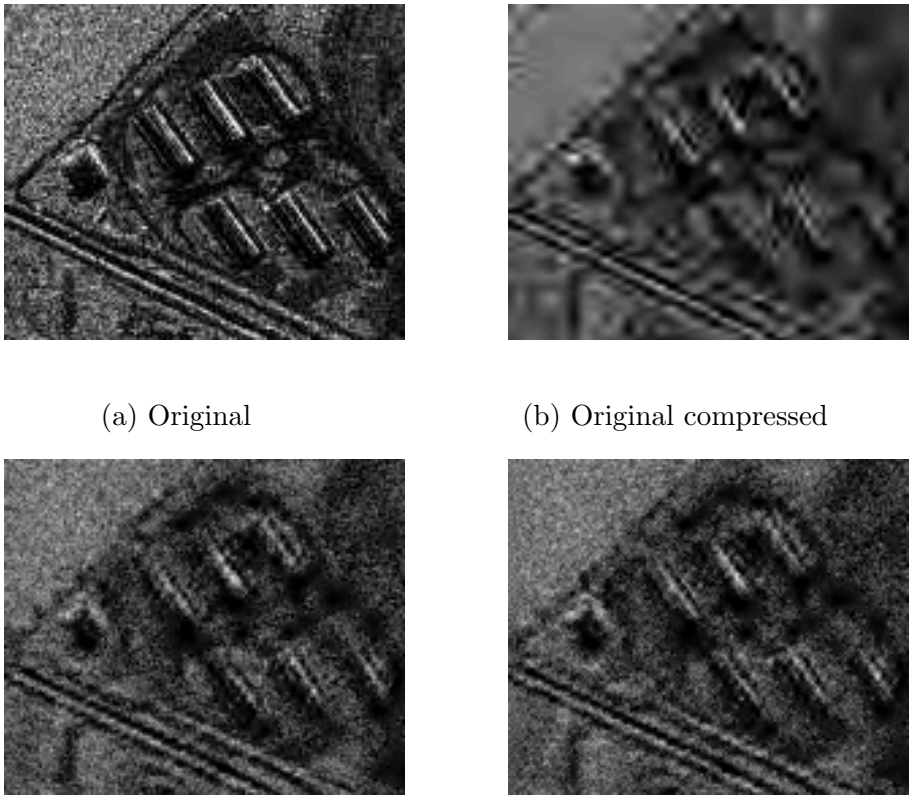
Figure 86: Another region from horse track image at 0.25 bpp



(a) Original (b) Original compressed

(c) AR model (d) Heeger-Bergen model

Figure 87: Detail from horse track image at 0.08 bpp



(a) Original

(b) Original compressed

(c) AR model

(d) Heeger-Bergen model

Figure 88: Another region from horse track image at 0.08 bpp

CHAPTER VII

CONCLUSIONS

7.1 Contributions

The contributions of this thesis are in two fronts.

In one front, we have extended the theory of the Bamberger DFB in three ways. First we identified the use of ladder structures to achieve perfect reconstruction FIR implementations that have an efficient implementation in the polyphase domain. Previously, the implementation proposed by Bamberger only provided PR for a particular family of IIR filters, and non-trivial FIR solutions only satisfied the alias-cancellation condition. Our implementation inherits all the nice properties of ladder structures, including flexible filter design and design parameter control, non-linear transformations, etc.

Second, we introduced the undecimated directional filter bank (UDFB). This is an undecimated version of Bamberger's DFB which retains the efficient polyphase domain implementation. The UDFB is achieved by manipulation of the resampling operations and proper use of multirate identities. The fan filter bank structure is modified such that the rotation-downsampling properties of the quincunx matrices are decoupled into separate operations. The main feature of the UDFB is that it has the shift invariance property.

Third, we proposed different polar-separable directional pyramids where the DFB or UDFB were combined with an image pyramid. We presented structures using the Gaussian pyramid, the Laplacian, and the radial component from the Steerable pyramid. The benefit of separating information directionally and across resolutions was higher than using a purely directional representation.

On a second front we have done an extensive evaluation of the DFB and UDFB-based decomposition for some important problems in image processing. Although there have been a couple of applications previously reported, mostly focused on target and character recognition, the DFB has been viewed as a "black box" and little work has been done

on understanding the impact that different parameters have on performance. As part of our evaluation we have conducted comparisons with competing image decompositions. In general we found that our results perform better or are competitive with other well known representations (e.g., Gabor filters, steerable pyramids, DWT and variants, local spatial transforms, etc.)

Hence, in the application arena we can cite the following contributions:

1. We implemented the commonly known texture classification algorithm using subband energy as features. We showed that the DFB was a top performer when compared with other representations. Its performance was very close to the TSWT and DST results reported in [19].

2. The directional nature of the data, allows us to generate rotationally invariant feature sets. We developed a set which included a pyramidal decomposition. Our results achieve similar or higher results over a test set when compared with the Gabor feature set of Haley and Manjunath [46]. We should not that the DFB-based system required much lower computational complexity, as well as the the feature calculation.

3. Following well known texture segmentation schemes, we compared our directional pyramids (DFB and UDFB) against the results reported by Randen [106] and De Rivaz [109]. The DFB-based systems on average outperformed the set of 11 multichannel transforms used in the comparison. The DT-CWT used by De Rivas came on a close second, which further validates that directional representations are better suited to this problem.

   We found that while the UDFB provides good boundary localization it comes with high storage cost. On the other hand, the DFB systems provided good region localization with poor boundaries at a significantly lower data rate. Hence, for analysis systems where good boundary delineation is not a must, the DFB based systems provide an excellent alternative.

4. We also evaluated the use of the DFB for texture synthesis by applying the Heeger-Bergen algorithm. The result show similar performance to the steerable pyramid. This

study was very simple, and further work with more complete texture models should follow. On the other hand, this model was good enough for the SAR compression algorithm of Chapter 6.

5. Following the advent of DWT denoising methods using subband coefficient thresholding, we applied the same principle using our directional pyramids. Although we did not see a big improvement in term of MSE, edges are preserved much better than the UDWT results. On the other hand the UDWT preserved better small features and showed less objectionable artifacts over smooths regions. The most objectionable artifacts for the UDFB-based system is the presence of cross hatching artifacts that reflect the shape of the 2-D impulse responses.

As a more specific application, we looked at SAR image processing. We presented speckle reduction techniques that show much better edge retention than the UDWT. We also showed that the segmentation system based on the DFB can provide good land cover identification. Finally, we presented a model-based compression system that uses texture models to capture the characteristics of speckle while the speckle free component is compressed at very low bitrates with a wavelet coder. At decompression, the speckle can synthesized from the texture model and added back to the scene if desired. The resulting images preserve much better visual quality.

7.2 Future Work

In this dissertation we have evaluated the DFB using well established algorithms for texture classification and segmentation, image denoising, etc. We found that the DFB and UDFB provide state-of-the-art results when compared to other image decompositions. Still, for the whole area of image analysis using multichannel transforms there is a long way to go in providing the best segmentation, denoising, etc. algorithms. Future work will be focussed on developing more advanced image analysis and understanding algorithms that use DFB- and UDFB-based pyramids as the front end processor.

In this thesis, we saw that better directional representation did not necessarily lead to better noise removal using subband coefficient thresholding. We saw that while the

directional pyramid provides better edge retention, it has more noticeable artifacts over smooth regions. On the other hand the UDWT has the opposite characteristics. As a next step we would like to study the combination of both schemes such that we obtain the best features of each and maximize visual quality.

For texture segmentation and classification, there are still unanswered issues. Ultimately, these operations are very simple tasks for humans, even under bad viewing conditions. It is clear that machines are far from achieving these tasks with ease. Our future objective will be to research the human visual system in order to design algorithms that efficiently exploit information across scales and resolutions. Beyond texture analysis, our ultimate objective will be to design a system capable to emulate our visual capabilities.

Finally we studied the analysis and enhancement of SAR images. Besides SAR there are other imaging technologies, e.g. hyper-spectral, infrared, etc., that produce volumes of images impossible to analyze by humans. It is necessary to develop automated systems that can analyze, understand, classify and index these image sets. We have developed a set of basic tools which could become the basis for this type of information management systems.

APPENDIX A

IMPLEMENTATION OF PARALLELOGRAM FILTER BANKS WITH THE DIAMOND FILTER BANK.

This appendix shows how a parallelogram filter bank can be implemented with a unimodular matrix followed by a diamond filter bank with quincunx decimation matrix $\mathbf{Q}$. This is an attractive feature, as it simplifies the implementation of the Bamberger DFB to one structure combined with the proper unimodular resampling operations.

We start with the parallelogram filter bank of Figure 89. In this discussion, we assume the filters $P_0(\omega)$ and $P_1(\omega)$ have the ideal parallelogram supports Ω_0 and Ω_1 respectively, as presented in Figure 90-(a).

The downsampling matrix $\mathbf{P}$ gives alias-free maximally decimated outputs. We can easily derive $\mathbf{P}$ by looking at the support of the Ω_0. By inspection we have

$$-\pi \leq 2\omega_0 - \omega_1 < \pi \cap -\pi \leq \omega_1 < \pi. \tag{89}$$

from which it is straightforward to obtain (see [130])

$$\mathbf{P} = \begin{bmatrix} 0 & -2 \\ 1 & 1 \end{bmatrix}. \tag{90}$$

Since $|\det \mathbf{P}| = |\det \mathbf{Q}| = 2$, then we can find a unimodular matrix, $\mathbf{U}$ such that $\mathbf{P} = \mathbf{UQ}$. We can next substitute $\mathbf{P}$ in Figure 89 with a series cascade of the resampling

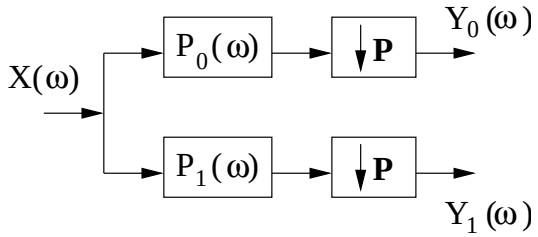


Figure 89: Two-channel parallelogram filter bank

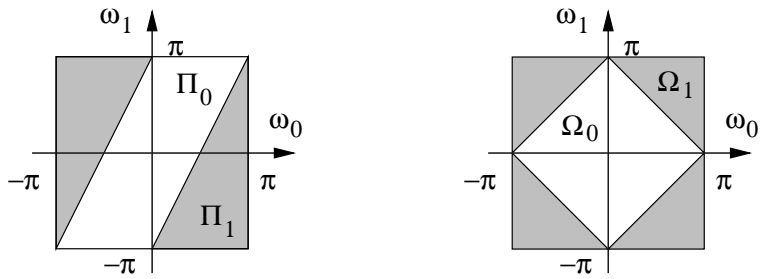


Figure 90: Ideal frequency domain supports for (a) parallelogram filter bank, (b) diamond filter bank

matrices **U** and **Q** as shown in Figure 91-(a). Applying the multirate identity from Figure 92, we can rearrange the structure as in Figure 91-(b).(We note that to apply this identity we have assumed the **U** is performing an upsampling operation, which in this case will be upsampling by a factor of one.)

It remains to show that the filters $P_0(\mathbf{U}^T\omega)$ and $P_1(\mathbf{U}^T\omega)$ have complementary diamond support. For this we let

$$\mathbf{Q} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}, \tag{91}$$

from which we find

$$\mathbf{U} = \mathbf{P}\mathbf{Q}^{-1} = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}. \tag{92}$$

Evaluating the unimodular transformation over the set of points in the Π_0 region, we see that the support of $P_0(\omega)$ gets transformed into

$$-\pi \leq \omega_0 + \omega_1 < \pi \cap -\pi \leq \omega_1 - \omega_0 < \pi, \tag{93}$$

which is the diamond support Ω_0 in Figure 90-(b). Similarly for $P_1(\mathbf{U}^T\omega)$, we obtain the diamond complementary Ω_1 region. Hence the equivalence between both filter bank structures is verified. We show the final two-channel structure in Figure 91-(c) obtained by letting $H_0(\omega) = P_0(\omega)$ and $H_1(\omega) = P_1(\omega)$, and moving **U** to the front of the filter bank.

We should note that the value of **B** and hence **U** depends on the parallelogram support. As shown in Chapter 1, there are other parallelogram partitionings of the frequency plane for which corresponding **U** exist.

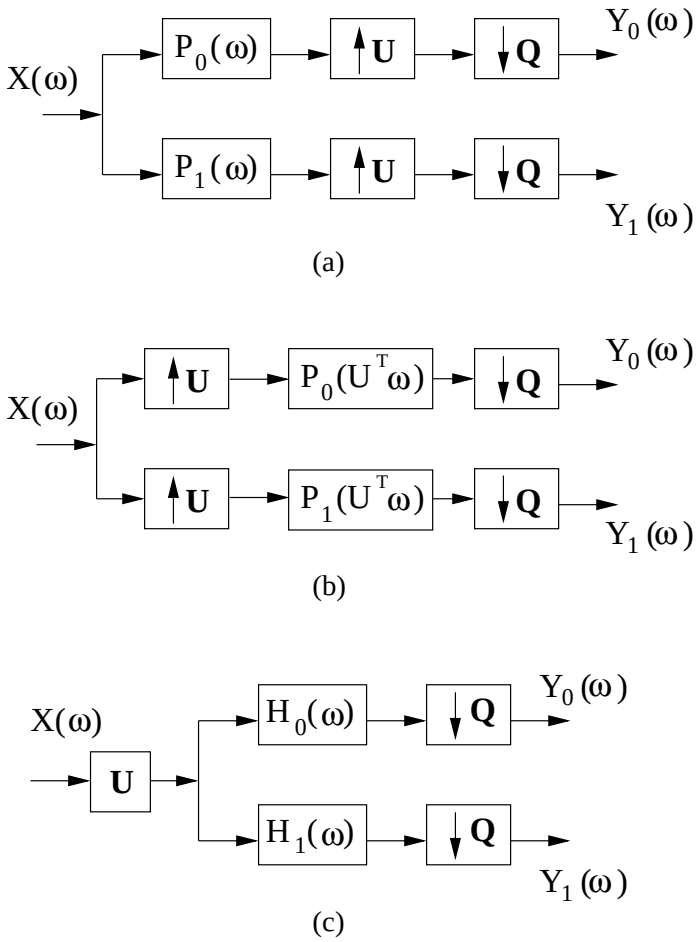


Figure 91: Transitions steps to implement a parallelogram filter bank with a diamond filter bank.

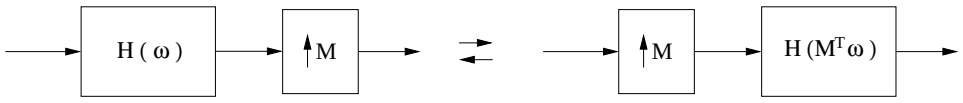


Figure 92: Filter-upsampler multirate identity.

We have focussed in the analysis stage, but the equivalent relationship for the synthesis stage is straightforward to derive by following similar steps.

REFERENCES

[1] Achim, A., Tsakalides, P., and Bezerianos, A., "Sar image denoising via bayesian wavelet shrinkage based on heavy-tailed modeling," *IEEE Trans. Geoscience and Remote Sensing*, vol. 41, pp. 1773–1784, August 2003.

[2] Ansari, R., "Efficient iir and fir fan filters," *IEEE Trans. on Cir. and Syst.*, vol. CAS–34, pp. 941–945, Aug. 1987.

[3] Ansari, R., Kin, C. W., and Dedovic, M., "Structure and design of two–channel filter banks derived from a triplet of halfband filters," *IEEE Transactions on Circ. Sys.–II: Anal. and Dig. Sig. Proc.*, vol. 46, pp. 1487–1496, December 1999.

[4] Arsenault, H. H. and April, G., "Properties of speckle integrated with a finite aperture and logarithmically transformed," *J. Opt. Soc. Am.*, vol. 66, pp. 1160–1163, November 1976.

[5] Bamberger, R. H., *The Directional Filterbank: A Multirate Filter Bank for the Directional Decomposition of Images*. PhD thesis, Georgia Institute of Technology, November 1990.

[6] Bamberger, R. H., "New results on two and three dimensional directional filter banks," in *Twenty-Seventh Asilomar Conference on Signals, Systems an Computers*, vol. 2, pp. 1286–1290, 1993.

[7] Bamberger, R. H. and Smith, M. J. T., "A multirate filter bank based approach for the detection and enhancement of linear features in images," in *Proceeedings of the 1991 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2557–2560, 1991.

[8] Bamberger, R. H. and Smith, M. J. T., "A filter bank for the directional decomposition of images," *IEEE Transactions on Signal Processing*, vol. 40, pp. 882–893, April 1992.

[9] Baxter, R. A., "Sar image compression with the gabor transform," *IEEE Trans. Geoscience and Remote Sensing*, vol. 37, pp. 574–588, January 1999.

[10] Besag, J. E., "Spatial interaction and statistical analysis of lattice systems," *J. Roy. Stat. Soc. series B*, vol. B-36, pp. 192–236, 1974.

[11] Bovik, A. C., Clark, M., , and Geisler, W. S., "Multichannel texture analysis using localized spatial filters," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 55–73, 1990.

[12] Brodatz, P., *Textures, A photographic album for artists and designers*. New York: Dover Publications, 1966.

[13] Bruekers, A. M. and ven den Enden, A. W. M., "New networks for perfect inversion and perfect reconstruction," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 130–137, January 1992.

[14] Burt, P. J. and Adelson, E. H., "The lapacian pyramid as a compact image code," *IEEE Trans. Communications*, vol. COM-31, pp. 532–540, April 1983.

[15] Candes, E. J. and Donoho, D. L., "Curvelets and curvilinear integrals," tech. rep., Stanford University, Dept. of Statistics, December 1999.

[16] Chang, C. Y., Kwok, R., and Curlander, J. C., "Spatial compression of seasat sar imagery.," *IEEE Trans. Geoscience and Remote Sensing*, vol. 26, no. 5, pp. 673–685, 1988.

[17] Chang, S. G., Yu, B., and Vetterli, M., "Image denoising via lossy compression and wavelet thresholding," *IEEE Transactions on Image Processing*, vol. 9, pp. 1532–1546, Sept. 2000.

[18] Chang, S. G., Yu, B., and Vetterli, M., "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Transactions on Image Processing*, vol. 9, pp. 1522–1531, Sept. 2000.

[19] Chang, T. and Kuo, C.-C. J., "Texture analysis and classification with tree-structured wavelet transform," *IEEE Transactions on Image Processing*, vol. 2, no. 4, pp. 429–441, 1993.

[20] Chen, J.-L. and Kundu, A., "Rotation and grayscale transform invariant identification using wavelet decomposition and hidden markov model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16, Feb. 1994.

[21] Cohen, F. S., Fan, Z., and Patel, M. A., "Classification of rotated and scaled textured image using gaussian-markov random field models," *ieeepami*, vol. 13, pp. 192–202, Feb. 1991.

[22] Comer, M. L. and Delp, E. J., "Segmentation of textured images using a multiresolution gaussian autoregressive model," *IEEE Transactions on Image Processing*, vol. 8, pp. 408–420, March 1999.

[23] Conners, R. W. and Harlow, C. A., "Toward a structural texture analyzer based on statistical methods," *Computer Graphics and Image Processing*, vol. 12, pp. 224–256, 1980.

[24] Crochiere, R. E., "Digital ladder structures and coefficient sensitivity.," October 1972.

[25] Cross, G. R. and Jain, A. K., "Markov random field texture model," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 25–39, jan 1983.

[26] Curlander, J. C. and McDonough, R. N., *Synthetic aperture radar : systems and signal processing / John C. Curlander, Robert N. McDonough.* Wiley series in remote sensing., New York : Wiley, c1991., 1991. A Wiley-Interscience publication. Includes bibliographical references and index.

[27] Daubechies, I. and Sweldens, W., "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245–267, 1998.

[28] Daugman, J. G., "Two-dimensional spectral analysis of cortical receptive field profile," *Vision Research*, vol. 20, pp. 847–856, 1980.

[29] Daugman, J. G., "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Am. A*, vol. 2, pp. 1160–1169, July 1985.

[30] Delp, E., Kashyap, R., and Mitchell, O., "Image data compression using autoregressive time series models," *Pattern Recognition*, vol. 11, pp. 313–323, dec 1979.

[31] Docef, A. and Smith, M., "A robust model-based coding technique for ultrasound video," in *Proceeedings of the 1995 SPIE Medical Imaging Conference*, pp. 203–213, February 26 – March 2, 1995. San Diego, CA.

[32] Donoho, D., "De-noising via soft-thresholding," tech. rep., Standford University, 1992.

[33] Donoho, D. L. and Johnstone, I. M., "Ideal spatial adaptation via wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[34] Eichel, P. and Ives, R. W., "Compression of complex-valued sar images," *IEEE Trans. Image Processing*, vol. 8, pp. 1483–1487, October 1999.

[35] Fatemi-Ghomi, N., Petrou, M., and Palmer, P. L., "Performance evaluation of texture segmentation algorithms based on wavelets," in *Proceedings of the workshop on Performance Characteristics of Vision Algorithms ECCV-96*, April 1996.

[36] Foucher, S., Benie, G., and Boucher, J.-M., "Multiscale map filtering of sar images," *IEEE Transactions on Image Processing*, vol. 10, pp. 49–60, Jan. 2001.

[37] Francos, J. M. and Meiri, A. Z., "A 2-d autoregressive, finite support causal model for texture analysis and synthesis," in *IEEE Proc. Int. Conf. on Acous., Speech, and Sig. Proc.*, pp. 1552–1555, 1989.

[38] Francos, J. M., Meiri, A. Z., and Porat, B., "A unified texture model based on a 2-d wold like decomposition," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2665–2678, January 1993.

[39] Frost, V., Stiles, J., Shanmugan, K., and Holtzman, J., "A model for radar images and its application to adaptive digital filtering of multiplicative noise," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 157–166, Mar. 1982.

[40] Fukuda, S. and Hirosawa, H., "Supression of speckle in synthetic aperture radar images using wavelet," *Int. J. Remote Sensing*, vol. 19, no. 3, pp. 507–519, 1998.

[41] Gagalowicz, A. and Ma, S., "Sequential synthesis of natural textures," *Computer Graphics and Image Processing*, vol. 30, pp. 289–315, Jan. 1985.

[42] Gagnon, L. and Jouan, A., "Speckle filtering of SAR images - a comparative study between complex-wavelet-based and standard filters," in *Proc. SPIE, Wavelet Applications in Signal and Image Processing V*, no. 3169, pp. 80–91, 1997.

[43] Gonzalez, J. G., Smith, M. J. T., Hontsch, I., Karam, L., Namuduri, K., and Szu, H., "Perceptual image compression for data tansmission in the battlefield," in *Visual Information Processing VII, Aerosense'98*, vol. SPIE-3387, April 1998. Orlando, Florida.

[44] Greenspan, H., Belongie, S., Goodman, R., and Perona, P., "Rotation invariant texture recognition using a steerable pyramid," in *IEEE Proc. Int. Conf. on Image Proc.*, (Jerusalem, Israel), Oct. 1994.

[45] Haley, G. M. and Manjunath, B. S., "Rotation invariant texture classification using modified gabor filters," in *IEEE Proc. Int. Conf. on Image Proc.*, (Washington, DC), Oct. 1995.

[46] Haley, G. M. and Manjunath, B. S., "Rotation-invariant texture classification using a complete space-frequency model," *IEEE Transactions on Image Processing*, vol. 2, pp. 255–269, feb 1999.

[47] Haralick, R. M., "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, pp. 786–804, May 1979.

[48] Hassner, M. and Sklansky, J., "The use of random fields as models of texture," *Computer Graphics and Image Processing*, vol. 12, pp. 357–370, Apr. 1980.

[49] Heeger, D. J. and Bergen, J. R., "Pyramid-based texture analysis/synthesis," in *Computer Graphics Proceeedings*, pp. 229–238, 1995.

[50] Hervet, E., Fjortoft, R., Marthon, P., and Lopes, A., "Comparison of wavelet-based and statistical speckle filters," in *EUROPTO Conference on SAR Image Analysis*, Sept. 1998. Barcelona, Spain.

[51] Hill, P. R., Bull, D. R., and Canagarajah, C. N., "Rotationally invariant texture features using the dual-tree complex wavelet transform," in *IEEE Proc. Int. Conf. on Image Proc.*, September, Vancouver, Canada 2000.

[52] Hoekman, D. H., "Speckle ensemble statistics of logarithmically scaled data," *IEEE Trans. Geoscience and Remote Sensing*, vol. 29, pp. 180–182, January 1991.

[53] Iversen, H. and nnestad, T. L., "An evaluation of stochastic models for analysis and synthesis of gray-scale texture," *Pattern Recognition Letters*, vol. 15, pp. 575–585, June 1994.

[54] Jain, A. and Zongker, D., "Feature selection: Evaluation, application and small sample performance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[55] Jain, A. and Zongker, D., "Feature selection: Evaluation, application and small sample performance," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[56] Jain, A. K. and Farrokhnia, F., "Unsupervised texture segmentation using gabor filters.," *Pattern Recognition*, vol. 24, no. 12, 1991.

[57] Jakowatz, C. V., *Spotlight-mode synthetic aperture radar : a signal processing approach.* Boston : Kluwer Academic Publishers, c1996., 1996. Includes bibliographical references and index.

[58] Jansen, M. and Roose, D., "Bayesian correction of wavelet threshold procedures for image de-noising," in *Joint Statistical Meetings*, August 1997. Anaheim, CA, USA.

[59] Johnston, J. D., "A filter family designed for use in quadrature mirror filter banks," in *Proc. IEEE Int. Conf. Acoust. Speech and Signal Proc.*, pp. 291–294, April 1980.

[60] Johnstone, I. M. and Silverman, B. W., "Wavelet threshold estimators with corelated noise," *Journal of the Royal Statistical Society, Series B*, vol. 59, pp. 319–351, 1997.

[61] Jónsson, R. H., *Adaptive Subband Coding of Video Using Probability Distribution Models.* PhD thesis, Georgia Institute of Technology, May 1994.

[62] Jónsson, R. H. and Mersereau, R. M., "Some properties of signals with generalized gaussian distribution with applications to image coding," in *Proceeedings of the 1994 Princeton Conference on Information Science and Systems*, 1994.

[63] Julez, B., "Visual pattern discrimination," *RE Trans. on Information Theory*, vol. 232, pp. 84–92, Feb. 1962.

[64] Julez, B., "Experiments in the visual perception of texture," *Sci. Amer.*, pp. 34–43, Apr. 1975.

[65] Kalker, C. C. M. and Shah, I. A., "Ladder structures for multidimensional linear phase perfect recosntruction filter banks and wavelets," in *SPIE Visual Communications and Image Processing'92*, vol. 1818, pp. 2–20, 1992.

[66] Kashyap, R. L. and Chellapa, R., "Estimation and choice of neighbors in spatial-interaction models of images," *IEEE Trans. on Information Theory*, vol. 29, pp. 60–72, Jan. 1983.

[67] Kashyap, R. L. and Chellapa, R., "Texture synthesis using noncausal autoregressive models," *IEEE Trans. Acous., Speech, and Sig. Proc.*, vol. ASSP-33, pp. 194–203, Jan. 1985.

[68] Kashyap, R. L. and Khotanzad, A., "A model-based method for rotation invariant texture classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 472–481, July 1986.

[69] Kingsbury, N. G., "The dual-tree complex wavelet transform: A new efficient tool for image restoration and enhancement," in *Proceedings of the European Signal Processing Conference, EUSIPCO 98*, vol. 1, pp. 319–322, September 1998. Rhodes.

[70] Kohonen, T., "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, Sept. 1990.

[71] Kuan, D., Sawchuck, A., Strand, T., and Chavel, P., "Adaptive noise smoothing filter for images with signal dependent noise," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 7, pp. 165–177, Mar. 1985.

[72] Laws, K. I., "Rapid texture identification," in *Proc. SPIE*, vol. 238, pp. 376–380, 1980.

[73] Lee, J. S., "Digital image enhancement and noise filtering by use of local statistics," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 2, pp. 165–168, Mar. 1980.

[74] Leung, M. M. and Peterson, A. M., "Scale and rotation invariant texture classification," in *Proc. 26th Asilomar Conf. Signals, Systems and Computers*, (Pacific Grove, CA), Oct. 1992.

[75] Lin, Y.-C., Chang, T., and Kuo, C.-C. J., "Hierarchical texture segmentation with wavelet packet transform and adaptive smoothing," in *Visual communications and image processing '94*, vol. 2308, September 1994. Chicago, Illinios.

[76] Liu, F. and Picard, R. W., "Periodicity, directionality and randomness: Wold features for image modeling and retrieval," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 722–733, July 1996.

[77] Lopes, A., Nezry, E., Touzi, R., and Laur, H., "Maximum a posteriori speckle filtering and first order texture models in sar imaging," in *IGARSS'90*, pp. 2409–2412, Sept. 1990.

[78] Lu, S. Y. and Fu, K. S., "Stochastic tree grammar inference for texture synthesis and discrimination," *Computer Graphics and Image Processing*, vol. 9, pp. 234–245, 1979.

[79] "HUT–CIS research SOM–PAK, LVQ–PAK." http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml.

[80] Man, H., Kossentini, F., and Smith, M. J. T., "A family of efficient and channel error resilient wavelet/subband image," *IEEE Trans. on Cir. Syst. Video Tech.*, vol. 9, pp. 95–108, February 1999.

[81] Mao, J. and Jain, A., "Texture classification and segementation using multiresolution simultaneous autoregressive models," *pr*, vol. 25, no. 2, pp. 173–188, 1992.

[82] Mascarenhas, N. D. A., "An overview of speckle noise filtering in SAR images," in *Primeras jornadas latinoamericanas de percepcin remota por radar: tcnicas de processamento de imgenes*, pp. 71–79, 1996. Argentina.

[83] Matsuyama, T., Miura, S., and Nagao, M., "Structural analysis of natural textures by fourier transformation," *Computer Vision, Graphics, and Image Processing*, vol. 24, pp. 347–362, 1983.

[84] Matsuyama, T., Saburi, K., and Nagao, M., "A structural analysis for regularly arranged textures," *Computer Graphics and Image Processing*, vol. 18, pp. 259–278, 1982.

[85] Mayer, H. A. and Somol, P., "Conventional and evolutionary feature selection of sar data using a filter approach," in *Systematics, Cybernetics and Informatics 2000*, July 2000. Orlando,FL.

[86] Meyer, F. G. and Coifman, R. R., "Brushlets: a tool for directional image analysis and image compression," *Applied and Computational Harmonic Analysis*, vol. 4, pp. 147–187, 1997.

[87] Mihcak, M. K., Kozintsev, I., and Ramchandran, K., "Spatially adaptive statistical modeling of wavelet image coefficients and its application to denoising," in *Proceeedings of the IEEE International Conference on Image Processing*, pp. 3253–3256, March 1999.

[88] Mitra, Sanjit K. Sherwood, R. J., "Digital ladder networks.," *IEEE Trans Audio Electroacoust*, vol. Au-21, pp. 30–36, February 1973.

[89] "MSTAR SAR Database." http://www.mbvlab.wpafb.af.mil/public/sdms/datasets/mstar/overvi

[90] Odegard, J. E., Guo, H., Lang, M., Burrus, C. S., Jr., R. O. W., Nowak, L. M., and Hiett, M., "Wavelet based sar speckle reduction and image processing," in *Proceedings of the SPIE, Algorithms for Synthetic Aperture Radar Imagery II*, April 1995.

[91] Ojala, T., Pietikäinen, M., and Mäenpää, T., "Gray scale and rotation invariant texture classification with local binary patterns," in *Proc. Sixth European Conference on Computer Vision*, no. 1, (Dublin, Ireland), pp. 404–420, 2000.

[92] Oliver, C. J., "Information from SAR images," *J. Phys. D: Appl. Phys.*, vol. 24, pp. 1493–1514, 1991.

[93] Oliver, C. and Quegan, S., *Understanding Aperture Radar Images*. Artech House Publishers, 1998.

[94] Owens, J. W. and Marcellin, M. W., "Rate allocation for spotlight sar phase history data compression," *IEEE Trans. on Image Processing*, vol. 8, pp. 1527–1533, November 1999.

[95] Park, S.-I., *New Directional Filter Banks and their Application in Image Processing*. PhD thesis, Georgia Institute of Technology, November 1999.

[96] Park, S.-I., Smith, M. J. T., and Mersereau, R. M., "A new directional filter bank for image analysis and classification," in *Proceeedings of the 1999 IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 1286–1290, October 1999. Chicago, IL.

[97] Parkes, S. M. and Clifton, H. L., "The compression of raw SAR and SAR image data," *Int. J. Remote Sensing*, vol. 20, no. 18, pp. 3563–3581, 1999.

[98] Phoong, S.-M., Kin, C. W., Vidyanathan, P. P., and Ansari, R., "A new class of two-channel biorthogonal filter banks and wavelet bases," *IEEE Transactions on Signal Processing*, vol. 43, pp. 649–665, March 1995.

[99] Pizurica, A., Philips, W., Lemahieu, I., and Acheroy, M., "Image de-noising in the wavelet domain using prior spatial constraints," in *Proc. of the IEE Int. Conf. on Image Proc. and its App.*, pp. 216–219, July 1999. Manchester, UK.

[100] Pizurica, A., Philips, W., Lemahieu, I., and Acheroy, M., "Despeckling SAR images using wavelets and a new class of adaptive shrinkage estimators," in *IEEE Proc. Int. Conf. on Image Proc.*, pp. II: 233–236, Oct. 2001. Thessaloniki, Greece.

[101] Porter, R. and Canagarajah, C. N., "Rotation invariant texture classification schemes using gmrfs and wavelets," in *Proceeedings of the International Workshop on Image and Signal Processing*, pp. 183–186, Nov. 1996.

[102] Porter, R. and Canagarajah, N., "A robust automatic clustering scheme for image segmentation using wavelets," *IEEE Transactions on Image Processing*, vol. 5, no. 4, pp. 662–665, 1996.

[103] Portilla, J. and Simoncelli, E. P., "Texture modeling and synthesis using joint statistics of complex wavelet coefficients," in *Proceedings of the IEEE Workshop on Statistical and Computational Theories of Vision*, June 1999. Fort Collins, CO.

[104] Quegan, S., "Segmentation and classification in SAR imagery," in *Primeras jornadas latinoamericanas de percepcin remota por radar: tcnicas de processamento de imgenes*, pp. 141–151, 1996. Argentina.

[105] "Randen-Husøy Texture Segmentation Framework." http://www.ux.his.no/sigproc/, 1999.

[106] Randen, T. and Husøy, J. H., "Filtering for texture classification: A comparative study," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, apr 1999.

[107] Randen, T. and Husøy, J. H., "Texture segmentation using filters with optimized energy separation," *IEEE Transactions on Image Processing*, vol. 8, apr 1999.

[108] Randolph, J. R. and Smith, M. J. T., "A directional reprsentation for binary images," in *ICASSP*, May 2001.

[109] Rivaz, P. D., *Complex Wavelet Based Image Analysis and Synthesis.* PhD thesis, University of Cambridge, 2000.

[110] Rosiles, J. G. and Smith, M. J. T., "Image denoising using directional filter banks," in *Proceeedings of the IEEE International Conference on Image Processing*, September, Vancouver, Canada 2000.

[111] Rosiles, J. G. and Smith, M. J. T., "Texture segmentation using a biorthogonal directional decomposition," in *Systematics, Cybernetics and Informatics 2000*, July 2000. Orlando,FL.

[112] Rosiles, J.-G. and Smith, M. J. T., "Compression of sar and ultrasound imagery using texture models," in *Proceedings of the Data Compression Conference*, April 1999. Snonwbird, Utah.

[113] Rosiles, J.-G. and Smith, M. J. T., "Model-based coding of sar and ultrasound images," in *IEEE ICECS 2000*, vol. 1, pp. 62–65, Dec. 2000.

[114] Ryan, T. W., Darwin, L. D., Fisher, H. D., and Iverson, A. E., "Image compression by texture modeling in the wavelet domain," *IEEE Trans. on Image Processing*, vol. 5, pp. 26–36, January 1996.

[115] "Sandia Laboratories SAR website." http://www.sandia.gov/RADAR/sar.html.

[116] Sedlmeyer, A. and Fettweis, A., "Digital filters with true ladder configuration.," *Int J Circuit Theory Appl*, vol. 1, pp. 5–10, March 1973.

[117] Simard, M., DeGrandi, G., Thomson, K. P. B., and B.Benie, G., "Analysis of speckle noise contribution on wavelet decomposition of sar images," *IEEE Trans. Geoscience and Remote Sensing*, vol. 36, pp. 1953–1962, November 1998.

[118] Simoncelli, E. P., Freeman, W. T., Adelson, E. H., and Heeger, D. J., "Shiftable multi-scale transforms," *IEEE Trans. Information Theory Special Issue on Wavelets*, vol. 38, pp. 587–607, March 1992.

[119] "Signal and Image Processing Inst., Univ. Southern California." http://www.sipi.usc.edu.

[120] Smith, M. J. and Eddins, S. L., "Analysis/synthesis techniques for subband image coding," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, pp. 1446–1456, August 1990.

[121] Tamura, H., Mori, S., and Yamawaki, T., "Textural features corresponding to visual perception," *IEEE Trans. on Syst., Man, Cybern.*, vol. SMC–8, pp. 460–473, June 1978.

[122] Tolhuizen, L., Hollmann, H., and Kalker, T. A. C. M., "On the realizability of biorthogonal, m-Dimensional two-band filter banks," *IEEE Transactions on Signal Processing*, vol. 43, pp. 640–648, March 1995.

[123] Tomita, T., Shirai, Y., and Tsuji, S., "Description of textures by a structural analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 183–191, Mar. 1982.

[124] Touzi, R., "A review of speckle filtering in the context of estimation theory," *IEEE Trans. Geoscience and Remote Sensing*, vol. 40, pp. 2392–2404, November 2002.

[125] Ulaby, F. T., Kouyate, F. K., Brisco, B., and Williams, T. H. L., "Textural information in SAR images," *IEEE Trans. Geoscience and Remote Sensing*, vol. GE-24, pp. 235–245, March 1986.

[126] Unser, M., "Local linear transforms for texture measurements," *Signal Processing*, vol. 11, pp. 61–79, jul 1986.

[127] Unser, M., "Texture discrimination using wavelets," *CVPR*, pp. 640–641, 1993.

[128] Unser, M., "Texture classification and segmentation using wavelet frames," *IEEE Transactions on Image Processing*, vol. 4, pp. 1549–1560, Nov. 1995.

[129] Unser, M. and Eden, M., "Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering," *ieeesmc*, vol. 20, pp. 804–815, July/August 1990.

[130] Vaidyanathan, P., *Multirate Systems and Filter Banks*. Prentice Hall, 1993.

[131] Valois, R. L. D., Albrecht, D. G., and Thorell, L. G., "Spatial–frequency selectivity of cells in macaque visual cortex," *Vision Rs.*, vol. 22, pp. 545–559, 1982.

[132] Vautrot, P., Wouwer, G. V. D., and Scheunders, P., "Non–separable wavelets for rotation–invariant texture classification and segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 472–481, July 1998.

[133] Viscito, E. and Allebach, J. P., "The analysis and design of multidimensional fir perfect reconstruction filter banks for arbitrary sampling lattices," *IEEE Trans. on Cir. and Syst.*, vol. 38, pp. 29–41, Jan. 1991.

[134] Wei, D., Guo, H., Odegard, J. E., Lang, M., and Burrus, C. S., "Simultaneous speckle reduction and data compression using best wavelet packet bases with application to sar based atd/r," Tech. Rep. CML TR95-02, Rice University, 1995.

[135] Wilson, R. and Spann, M., "Finite prolate spheroidal sequences and their applications ii: Image feature description and segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 193–203, mar 1988.

[136] Wu, D. and Linders, J., "A new texture approach to discrimination of forest clearcut, canopy, and burned area using airborne C-band SAR," *IEEE Trans. Geoscience and Remote Sensing*, vol. 37, pp. 555–563, January 1999.

[137] Wu, W.-R. and Wei, S.-C., "Rotation and grayscale transform invariant texture classification using spiral resampling, subband decomposition, and hidden markov model," *IEEE Transactions on Image Processing*, vol. 5, Oct. 1996.

[138] Xie, H., Pierce, L., and Ulaby, F., "SAR speckle reduction using wavelet denoising and markov random field modeling," *IEEE Trans. Geoscience and Remote Sensing*, vol. 40, pp. 2196–2212, October 2002.

[139] Xie, H., Pierce, L., and Ulaby, F., "Statistical properties of logarithmically transformed speckle,," *IEEE Trans. Geoscience and Remote Sensing*, vol. 40, pp. 721–727, March 2002.

[140] You, J. and Cohen, H. A., "Classification and segmentation of rotated and scaled textured images using 'tuned' masks," *Pattern Recognition*, vol. 26, no. 2, pp. 245–258, 1993.

[141] Zeng, Z. and Cumming, I., "Bayesian speckle noise reduction using the discrete wavelet transform," in *IGARSS'98*, pp. 7–9, 1998.

[142] Zhu, S., Wu, Y., and Mumford, D., "Filters, random fields and maximum entropy (frame) - towards the unified theory for texture modeling," in *IEEE conference on Computer Vision and Pattern Recognition*, June 1996.

VITA

José Gerardo González Rosiles was born is Juárez, Mexico and came to the U.S. with his family in 1989. He did all of his pre-university education in Mexico obtaining several high honor recognitions. He graduated with Highest Honors from The University of Texas at El Paso (UTEP) with a Bachelor of Science in Electrical Engineering, and with a Master of Science in Electrical Engineering in the area of Digital Signal Processing. During his master's studies, he had Summer internship at the Jet Propulsion Laboratory where he worked on the compression of space imagery in the Galileo space probe project. He entered the Ph.D. Program at the Georgia Institute of Technology concentrating in the areas of Digital Signal Processing, Telecommunications, and Computer Engineering. During his stay at Georgia Tech he obtained a second MSEE degree with emphasis in computer engineering. His Ph.D. research was directed by Dr. Mark J. T. Smith. He focused in the development of directional filter banks and their application to image processing and texture analysis problems. During his Ph.D. studies he worked as an engineer for Iterated Systems, Inc. in the area of lossless image compression. From October 2000 to June 2004 he worked at Motorola Silicon Products Sector (now Freescale Semiconductors) in Austin Texas. He developed algorithms and architectures for digital imaging and digital video processing. Upon completion of his Ph.D., he joined the faculty of the Electrical and Computer Engineering Department at UTEP as an assistant professor. He is a member of the IEEE and Eta Kappa Nu. He plans to spend his days with his family, his wife Laura, his daughter Paulina and other future members of his family in the El Paso - Juárez region.